УДК 519.61, 519.852.2

# ИСПОЛЬЗОВАНИЕ ИДЕЙ АЛГОРИТМА QUICKHULL В МЕТОДЕ ДВОЙНОГО ОПИСАНИЯ

**С. И. Бастраков[1], Н. Ю. Золотых[1]**

Метод двойного описания, известный также как алгоритм Моцкина–Бургера, является одним из методов нахождения общего решения системы линейных неравенств. Предлагается его новая модификация с использованием идей алгоритма Quickhull. Приводятся результаты вычислительного эксперимента, показывающие превосходство предлагаемой модификации над оригинальным методом двойного описания и некоторыми его вариантами, а также — во многих случаях — и над алгоритмом Quickhull. Работа выполнена при финансовой поддержке РФФИ (код проекта 09–01–00545-а).

---

**Ключевые слова:** система линейных неравенств, выпуклая оболочка, конус, полиэдр, метод двойного описания, алгоритм Моцкина–Бургера.

**1. Введение.** Хорошо известно, что каждый многогранный конус в $\mathbb{R}^d$ может быть представлен любым из следующих двух способов:

— как множество решений некоторой однородной системы линейных неравенств;

— как множество всех неотрицательных комбинаций некоторой системы векторов.

Неприводимая система векторов, множество неотрицательных комбинаций которой есть заданный конус, называется его *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 определяет множество экстремальных лучей конуса.

В данной работе рассматривается задача построения остова многогранного конуса по заданному описанию в виде однородной системы линейных неравенств. Хорошо известно, что обратная задача не более, чем за линейное время, сводится к прямой. В случае полиэдров в $\mathbb{R}^d$ аналогом указанных задач являются задачи построения *вершинного/фасетного описания* — они легко сводятся к построению остова конуса в пространстве $\mathbb{R}^{d+1}$. Задача построения фасетного описания называется также задачей построения выпуклой оболочки.

Указанные задачи появляются во многих приложениях, например, в компьютерной графике, физической симуляции, обработке изображений, картографии, вычислительной биологии и многих других. К построению выпуклой оболочки могут быть сведены задачи построения триангуляции Делоне и диаграммы Вороного [1].

В [2] проведена классификация и анализ различных методов решения задачи, произведена оценка трудоемкости в зависимости от суммарного размера входа и выхода. Для всех известных классов алгоритмов построены параметрические семейства входных данных, трудоемкость работы на которых не ограничена сверху (в асимптотическом смысле с ростом $d$) никаким полиномом от суммарного размера входа и выхода. Вопрос о существовании полиномиальных в данном смысле алгоритмов остается открытым.

В настоящей статье предлагается новая модификация метода двойного описания [3], использующая идеи алгоритма Quickhull для построения выпуклой оболочки системы точек [4] и идеи метода, предложенного в [5]. Статья имеет следующую структуру: необходимые определения и обозначения приведены в разделе 2, раздел 3 посвящен методу двойного описания, раздел 4 — оригинальному алгоритму Quickhull, в разделе 5 содержится описание предлагаемой модификации, результаты вычислительных экспериментов приведены в разделе 6.

**2. Определения и обозначения.** *Многогранным конусом* в $\mathbb{R}^d$ (далее просто *конусом*) называется множество решений системы однородных линейных неравенств: $C = \{x \in \mathbb{R}^d : Ax \geqslant 0\}$, где $A \in \mathbb{R}^{m \times d}$. Говорят, что система $Ax \geqslant 0$ *определяет* конус $C$. Конус называется *острым*, если он не содержит ненулевых подпространств. Необходимым и достаточным условием остроты конуса является выполнение равенства $\operatorname{rank} A = d$. Любой многогранный конус может быть задан в виде конической оболочки конечной системы

[1] Нижегородский государственный университет им. Н. И. Лобачевского, факультет вычислительной математики и кибернетики, просп. Гагарина, 23, корп. 2, 603950, г. Нижний Новгород; С. И. Бастраков, магистрант, e-mail: Sergey.Bastrakov@gmail.com; Н. Ю. Золотых, доцент, e-mail: Nikolai.Zolotykh@gmail.com

векторов $u_1, u_2, \ldots, u_n$ в $\mathbb{R}^d$: $C = \{x = \alpha_1 u_1 + \alpha_2 u_2 + \ldots + \alpha_n u_n \, : \, \alpha_i \geqslant 0 \ (i = 1, 2, \ldots, n)\}$. 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, если для некоторого $\alpha > 0$ верно $u = \alpha v$. Луч $u \in C$ называется *экстремальным*, если из условий $u = \alpha v + \beta w$, $\alpha > 0$, $\beta > 0$, $v$, $w \in C$, следует, что $u$, $v$ и $w$ равны. Пусть $D$ — выпуклое множество в $\mathbb{R}^d$ и для некоторых $a \in \mathbb{R}^d$, $\alpha \geqslant 0$ выполнено $D \subseteq \{x : ax \leqslant \alpha\}$, тогда $D \cap \{x : ax = \alpha\}$ называется *гранью* множества $D$. 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 $U(C)$, множество всех пар $\{u, v\}$ смежных экстремальных лучей — через $E(C)$.

Выпуклая оболочка конечной системы векторов $v_1, v_2, \ldots, v_q$ в $\mathbb{R}^d$

$$P = \left\{ x = \beta_1 v_1 + \beta_2 v_2 + \ldots + \beta_q v_q \, : \, \beta_i \geqslant 0 \ (i = 1, 2, \ldots, q), \sum_{i=1}^{q} \beta_i = 1 \right\}$$

называется *политопом*. Любой политоп также может быть представлен в виде множества решений системы линейных неравенств: $P = \{x \in \mathbb{R}^d \, : \, Ax \leqslant b\}$. *Фасетой* политопа называется грань размерности $\dim P - 1$, где под $\dim P$ понимается размерность политопа, т.е. размерность минимального аффинного подпространства, содержащего $P$. *Гребнем* политопа называется грань размерности $\dim P - 2$, фасеты называют *смежными*, если их пересечение является гребнем. Под *графом фасет политопа* будем понимать граф, вершинами которого являются фасеты политопа, вершины графа смежны тогда и только тогда, когда смежны соответствующие фасеты. Выпуклая оболочка $d + 1$ аффинно независимой точки называется *$d$-мерным симплексом*. Политоп называется *симплициальным*, если все его фасеты являются симплексами соответствующей размерности.

**3. Метод двойного описания.** *Метод двойного описания* (*double description method*) [3], известный также как *алгоритм Моцкина–Бургера* [6], является одним из алгоритмов построения остова многогранного конуса. Существует множество его модификаций, например [5, 7, 8]. Описание ниже основано на варианте метода, предложенном в [5].

Далее рассматривается случай острого конуса, задача для произвольного конуса легко сводится к данной. На предварительном этапе выбирается квадратная невырожденная подматрица $A_d$ матрицы $A$. Остовом конуса, заданного подсистемой неравенств $A_d x \geqslant 0$, являются столбцы матрицы $A_d^{-1}$. Пусть известен остов $U$ конуса $C = \{x \in R^d \, : \, Ax \geqslant 0\}$, рассмотрим процедуру нахождения остова $U'$ конуса $C' = \{x \in R^d \, : \, Ax \geqslant 0, \ ax \geqslant 0\}$, $a \in \mathbb{R}^d$. Пусть $U_0 = \{u \in U \, : \, au = 0\}$, $U_+ = \{u \in U \, : \, au > 0\}$ и $U_- = \{u \in U \, : \, au < 0\}$. Тогда остов $U'$ конуса $C'$ определяется следующим образом: $U' = U_+ \cup U_0 \cup U_\pm$, где $U_\pm = \{w = (au)v - (av)u, \ u \in U_+, \ v \in U_-, \ \{u, v\} \in E(C)\}$.

Алгоритм состоит в последовательном добавлении всех неравенств исходной системы с перестройкой остова. Модификации отличаются друг от друга порядком добавления неравенств исходной системы, способом и временем определения смежности и представлением информации о смежности. На каждой итерации существенное время занимает построение (или перестроение) множества $E(C)$. Некоторые способы проверки смежности приведены в [5].

**4. Алгоритм Quickhull.** Quickhull [4] является известным алгоритмом построения выпуклой оболочки системы точек в $\mathbb{R}^d$. В данном разделе, следуя [4], мы приводим его неформальное описание для случая симплициального политопа размерности $d$. Заметим, что к алгоритму Quickhull близок другой алгоритм [9] построения выпуклой оболочки. Алгоритм [9] тоже использует триангуляции, но в нем отсутствует ряд усовершенствований из Quickhull.

Алгоритм является итеративным: построение выпуклой оболочки начинается с выбора набора точек, образующих симплекс, далее на каждой итерации выбирается очередная не принадлежащая ранее построенной выпуклой оболочке точка и происходит ее добавление. Для каждой из фасет хранится набор ее вершин, информация о смежности фасет представляется в виде списков смежности — такое представление задает граф фасет политопа в виде списков смежности.

С каждой из фасет ассоциируется *внешнее множество* — множество, состоящее из точек, лежащих с внешней стороны фасеты, т.е. не удовлетворяющих ее неравенству (фасета *видима* из точки). В случае, когда из точки видимы несколько фасет, она попадает во внешнее множество одной (любой) из них. В начале каждой итерации выбирается фасета с непустым внешним множеством и в качестве добавляемой точки выбирается наиболее удаленная из точек внешнего множества данной фасеты. Алгоритм завершает работу, когда внешние множества всех фасет пусты.

При добавлении очередной точки под *граничным гребнем* будем понимать гребень, являющийся пересечением смежных видимой и невидимой из добавляемой точки фасет. Перестроение выпуклой оболочки при добавлении выбранной точки происходит следующим образом.

**Алгоритм 1** (добавление новой точки в методе Quickhull [4]):

1) все фасеты делятся на видимые и невидимые из добавляемой точки;

2) граница области видимости является набором граничных гребней, из каждого граничного гребня создается новая фасета, проходящая через добавляемую точку;

3) точки из внешних множеств видимых фасет перераспределяются по внешним множествам новых фасет;

4) видимые фасеты удаляются;

5) обновляется информация о смежности.

Определение видимых фасет и границы области видимости производится при помощи поиска на графе фасет построенной к текущему моменту выпуклой оболочки, результатом поиска является список всех видимых фасет и список всех граничных гребней. Получение списка невидимых фасет в явном виде не производится, в симплициальном случае все фасеты, не попавшие в список видимых, являются невидимыми. Для удобства сформулируем алгоритм поиска на графе фасет в терминах фасет и их смежности, подразумевая соответствие между фасетами политопа и вершинами графа фасет. Список видимых фасет инициализируется фасетой, внешнему множеству которой принадлежала добавляемая точка (эта фасета является видимой по определению). На каждой итерации алгоритма поиска выбирается очередная фасета из списка найденных видимых и происходит обход всех ее не посещенных в процессе поиска смежных фасет. Если при этом найдена другая видимая фасета, то она добавляется в конец списка видимых, в противном случае найден граничный гребень, он добавляется в соответствующий список.

Точки из внешних множеств видимых фасет, не попавшие ни в одно из внешних множеств новых фасет при перераспределении, являются внутренними и исключаются из дальнейшего рассмотрения. Данная особенность алгоритма обусловливает его высокую эффективность в случае, когда среди входных точек есть много внутренних, в [4] приведены результаты подтверждающих это вычислительных экспериментов.

Каждая из новых фасет смежна ровно с одной из старых невидимых фасет (невидимой фасетой, образующей граничный гребень, из которого была создана новая фасета), остальные смежные с ней фасеты определяются из числа новых. Информация о смежности старых невидимых фасет друг с другом не меняется.

В [4] приведено описание алгоритма для симплициального случая, в общем случае предполагается триангуляция несимплициальных фасет. Такой подход существенно упрощает установление смежности между новыми фасетами, однако, как показано в [2], с ростом размерности число симплексов в такой триангуляции не ограничено сверху никаким полиномом от суммарного размера входа и выхода. В предлагаемой модификации метода двойного описания используется другой подход.

**5. Новая модификация метода двойного описания.** Предлагаемый алгоритм основан на модификации метода двойного описания [5] с использованием следующих особенностей алгоритма Quickhull:

— представление информации о смежности в виде списков смежности;

— использование поиска на графе;

— наличие внешних множеств и порядок добавления.

Далее приводится описание алгоритма в терминах построения остова многогранного конуса. Понятиям точки и фасеты выпуклой оболочки в описании Quickhull теперь соответствуют понятия неравенства и луча остова.

Для каждого построенного к текущему моменту луча остова $u \in U$ будем хранить список смежных лучей, набор таких списков определяет множество пар смежных лучей $E(C)$ (но в явном виде оно не хранится). Множество инцидентных лучу $u$ неравенств будем хранить в $Z(u)$, в приводимом ниже описании для удобства считается, что $Z(u)$ содержит соответствующие строки матрицы системы неравенств. Кроме того, для каждого луча будем хранить *внешнее множество* $\mathrm{OS}(u)$, состоящее из неравенств, которым он не удовлетворяет: если $a \in \mathrm{OS}(u)$, то $au < 0$, где $a$ — строка матрицы системы неравенств. В случае, когда неравенству не удовлетворяют несколько лучей, оно попадает во внешнее множество одного (любого) из них.

Алгоритм начинает работу с выбора базисной подсистемы неравенств и построения остова определяемого ей конуса согласно описанию в разделе 3. Далее на каждой итерации происходит добавление очередного неравенства системы, при этом также возможно исключение некоторых из еще не добавленных неравенств-следствий из дальнейшего рассмотрения. Порядок выбора добавляемого неравенства аналогичен используемому в Quickhull: выбирается любой луч $u$, такой, что $\mathrm{OS}(u) \neq \emptyset$, а в качестве очередного добавляемого неравенства выбирается $a = \arg\min\limits_{a \in \mathrm{OS}(u)} au$.

Пусть известен остов $U$ конуса $C$, рассмотрим процедуру добавления выбранного неравенства $ax \geqslant 0$, т.е. построения остова $U'$ конуса $C' = \{x \in C \,:\, ax \geqslant 0\}$.

**Алгоритм 2** (добавление нового неравенства):

1) строятся множества $U_+ = \{u \in U \,:\, au > 0\}$ , $U_- = \{u \in U \,:\, au < 0\}$ и $U_0 = \{u \in U \,:\, au = 0\}$; для всех $u_0 \in U_0$: $Z(u_0) := Z(u_0) \cup \{a\}$;

2) строятся новые лучи

$$U_\pm = \big\{w = (au)v - (av)u,\, u \in U_+,\, v \in U_-,\, \{u,v\} \in E(C)\big\}, \quad Z(w) = Z(u) \cap Z(v) \cup \{a\};$$

3) неравенства из внешних множеств лучей из $U_-$ перераспределяются по внешним множествам лучей из $U_\pm \cup U_0$;

4) формируется новый остов $U' = U_+ \cup U_0 \cup U_\pm$, лучи из $U_-$ удаляются;

5) обновляется информация о смежности.

Построение множеств $U_+$ , $U_-$ и $U_0$ производится при помощи поиска на графе способом, близким к используемому в Quickhull, $U_+$ и $U_-$ соответствуют невидимым и видимым фасетам в терминологии Quickhull. Наличие $U_0$ не приводит к существенному усложнению данной процедуры. В процессе поиска также определяются все пары $(u,v)$: $u \in U_+$, $v \in U_-$, $\{u,v\} \in E(C)$ ($E(C)$ не хранится в явном виде, но представление через списки смежности позволяет определять все такие пары способом, аналогичным используемому в Quickhull). Неравенства, не попавшие ни в одно из внешних множеств при перераспределении, являются следствиями и исключаются из дальнейшего рассмотрения.

При обновлении информации о смежности основным этапом является определение смежности лучей из $U_0 \cup U_\pm$. Оно может производиться одним из стандартных для метода двойного описания способов, например "графовой" модификацией "комбинаторного" правила, описанной в [5]. Проверка смежности является наиболее трудоемкой операцией алгоритма, при этом используются только данные об инцидентности лучей и неравенств, представленные в виде множеств $Z(u)$. При осуществлении тестов на смежность необходимо выполнение большого количества операций пересечения двух множеств, сами же $Z(u)$ меняются относительно редко (только в случае $u \in U_0$).

Рассмотрим варианты представления множеств $Z(u)$ в виде упорядоченных векторов и битовых полей. Первый способ используется в Quickhull и является наиболее естественным для триангулирующего алгоритма (в этом случае $\big|Z(u)\big|$ одинакова для всех $u$). Для предлагаемого метода он является предпочтительным, если каждому вектору инцидентно относительно малое количество неравенств. В этом случае проверка необходимого условия инцидентности лучей $u$ и $v$: $\big|Z(u) \cap Z(v)\big| \geqslant r - 2$, где $r = \operatorname{rank} A$, и достаточного условия в "графовой" модификации "комбинаторного" правила [5] может осуществляться относительно быстро. В методе двойного описания традиционно используется представление информации о смежности в неявном виде, через нулевые элементы матрицы $AU$, где $U$ — матрица, составленная из векторов построенного к текущему моменту остова. Данный способ практически эквивалентен представлению $Z(u)$ в виде битовых полей, в этом случае сложность выполнения операций для проверки условий смежности не зависит от количества инцидентных неравенств. На практике такое представление может оказаться предпочтительным в существенно вырожденных задачах. В разработанной компьютерной реализации используется первый способ.

**6. Вычислительный эксперимент.** Предложенный в настоящей работе алгоритм реализован в виде программы `qskeleton`.

Проведен вычислительный эксперимент, сравнивающий производительность программ `qskeleton`, `qhull` (реализация метода Quickhull, `http://www.qhull.org`) и программы `Skeleton` (реализация метода, предложенного в [5], `http://www.uic.unn.ru/~zny/skeleton`). Все вычисления проводились в арифметике с плавающей точкой двойной точности (`double`). В программе `Skeleton` использовался порядок добавления неравенств `maxcutoff` [5], остальные опции алгоритма — по умолчанию. Входные данные для экспериментов были получены программой `polyhedron` из пакета `Arageli` (`http://www.arageli.org`). Тестовая инфраструктура: Microsoft Windows 7, Intel Core i3 2.4 GHz, 4 GB RAM, Microsoft Visual C++ 2008 SP1.

Для удобства далее приводится описание экспериментов и классов входных данных в терминах задачи построения выпуклой оболочки (фасетного описания). Для применения `qskeleton` и `Skeleton` использовался стандартный способ сведения к задаче построения остова конуса; таким образом, все три используемые реализации решали эквивалентные задачи.

В первой серии экспериментов решалась задача построения выпуклой оболочки точек, равномерно распределенных в 6-мерном единичном шаре. В этом случае большое количество точек является внутренними и быстро исключается из дальнейшего рассмотрения в `qhull`, соответствующие им неравенства являются следствиями и быстро исключаются в `qskeleton`. Зависимость времени работы от количества точек приведена на рис. 1. На рис. 2 приведены результаты аналогичного эксперимента для распределенных на 6-мерной единичной сфере точек. В этом случае все точки являются вершинами выпуклой оболочки, однако большое преимущество `qskeleton` и особенно `qhull` над `Skeleton` сохраняется.

Это объясняется тем, что политоп является симплициальным и представление информации об инцидентности, используемое в `Skeleton`, неэффективно, как показано в конце раздела 5. Преимущество `qhull` над `qskeleton` обусловлено использованием ряда оптимизаций, специфических для триангулирующих алгоритмов. Тем не менее, использование идей Quickhull в данном случае позволило значительно улучшить производительность метода двойного описания.

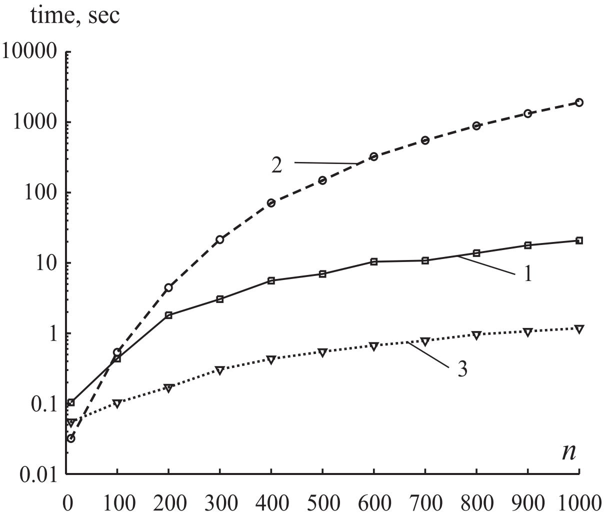


Рис. 1. Время построения выпуклой оболочки системы $n$ случайных точек, равномерно распределенных в 6-мерном шаре: 1) Qskeleton, 2) Skeleton, 3) Qhull

В [2] рассматриваются классы *“запутанных”* (*intricate*) политопов для задачи построения выпуклой оболочки. Они состоят из параметрических семейств политопов, число симплексов в любой триангуляции фасет которых неполиномиально от суммарного размера входа и выхода. Таким образом, алгоритмы, использующие триангуляцию фасет (и, в частности, Quickhull), не могут быть полиномиальными от суммарного размера входа и выхода на данном классе входных данных. В частности, в [2] приведены нижние оценки трудоемкости триангулирующих алгоритмов в зависимости от суммарного размера входа и выхода $s$ для следующих классов “запутанных” политопов: $\Omega\big(s^{(\log\log s)/4-1/2}\big)$ для класса гиперкубов $H_d$ и $\Omega\big(2^{\sqrt[3]{s}}\big)$ для класса произведений симплексов $TT_{2d}$.

На рис. 3 приведена зависимость времени работы от суммарного размера входа и выхода на классе $TT_{2d}$, состоящем из произведений двух $d$-мерных симплексов. На рис. 4 приведена зависимость времени работы от суммарного размера входа и выхода на классе гиперкубов. В отличие от Quickhull, метод двойного описания и предлагаемая его модификация полиномиальны от суммарного размера входа и выхода на данных классах.

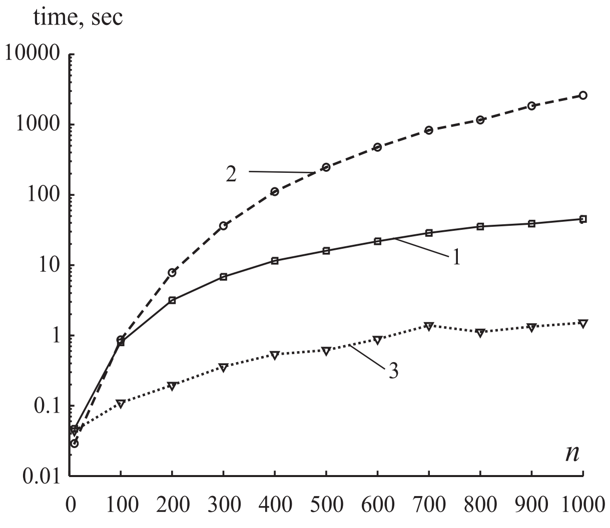


Рис. 2. Время построения выпуклой оболочки системы $n$ случайных точек, равномерно распределенных на 6-мерной сфере: 1) Qskeleton, 2) Skeleton, 3) Qhull

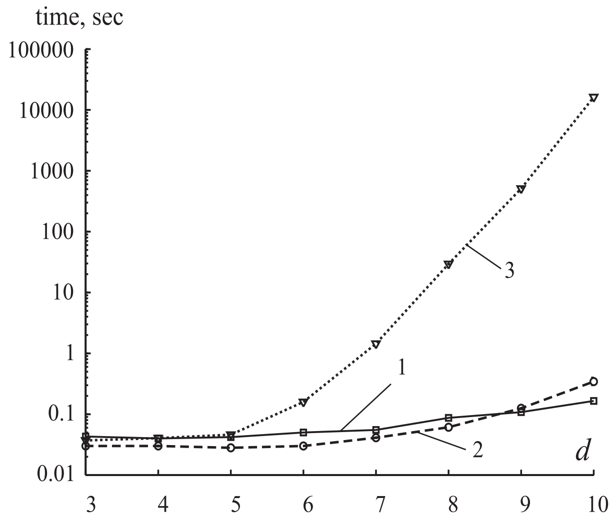


Рис. 3. Время построения фасетного описания произведения двух $d$-мерных симплексов: 1) Qskeleton, 2) Skeleton, 3) Qhull

Как показано в [2], семейство произведений циклических политопов $CC_{2\delta^2}(n)$ с ростом $\delta$ является примером класса входных данных, на котором все известные методы являются неполиномиальными от суммарного размера входа и выхода. На рис. 5 приведено время работы в зависимости от $n$ при $\delta = 2$ (при этом размерность пространства равна 8).

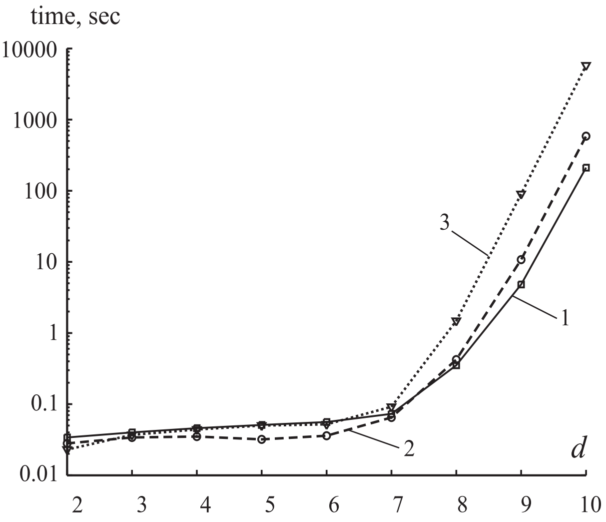


Рис. 4. Время построения фасетного описания $d$-мерного гиперкуба: 1) Qskeleton, 2) Skeleton, 3) Qhull

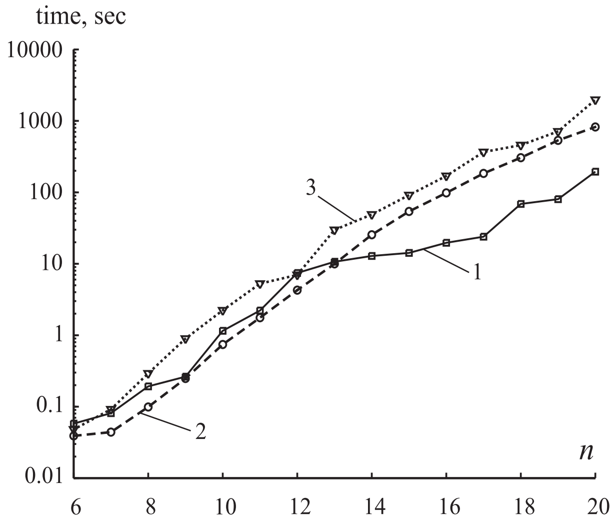


Рис. 5. Время построения фасетного описания для произведения двух 8-мерных $n$-вершинных циклических политопов: 1) Qskeleton, 2) Skeleton, 3) Qhull

**7. Заключение.** Предложена новая модификация метода двойного описания, использующая идеи алгоритма Quickhull. Приведены результаты вычислительных экспериментов, подтверждающих эффективность предложенной реализации.

Авторы благодарят В. Н. Шевченко и С. С. Лялина за полезные обсуждения.

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 // Computational Geometry. 1997. **2**, N 2. 265–301.
3. *Motzkin T., Raiffa H., Thompson G., Thrall R.M.* The double description method // Contributions to the Theory of Games. Princeton: Princeton University Press, 1953. 51–73.
4. *Barber C.B., Dobkin D.P., Huhdanpaa H.* The Quickhull algorithm for convex hulls // ACM Transactions on Mathematical Software. 1996. **22**, N 4. 469–483.
5. *Золотых Н.Ю.* Новая модификация метода двойного описания для построения остова многогранного конуса // Ж. вычисл. матем. и матем. физ. (направлено).
6. *Черников С.Н.* Линейные неравенства. М.: Наука, 1968.
7. *Шевченко В.Н., Груздев Д.В.* Модификация алгоритма Фурье–Моцкина для построения триангуляции // Дискр. анализ и исслед. операций. 2003. **10**, № 1. 53–64.
8. *Fukuda K., Prodon A.* Double description method revisited // Lecture Notes in Computer Science. Vol. 1120. Berlin: Springer-Verlag, 1996. 91–111.
9. *Черных О.Л.* Построение выпуклой оболочки конечного множества точек на основе триангуляции // Ж. вычисл. матем. и матем. физ. 1991. **31**, № 8. 1231–1242.

Поступила в редакцию
27.03.2011