УДК 519.852.2

АЛГОРИТМ МОЦКИНА-БУРГЕРА И ВЫЧИСЛЕНИЕ ВЫПУКЛЫХ ОБОЛОЧЕК ТОЧЕК N-МЕРНОГО ПРОСТРАНСТВА

П.А.Буровский*

Алгоритм Моцкина–Бургера вычисления всех решений системы линейных однородных неравенств позволяет построить алгоритм вычисления граней всех размерностей, а также их нормальных конусов, выпуклой оболочки конечного числа точек n-мерного пространства. В статье описывается реализация этих алгоритмов для системы компьютерной алгебры Maple в виде двух пакетов, `convexhull` *и* `motzkin_burger`*; приводятся примеры их использования.*

Алгоритм Моцкина–Бургера известен достаточно давно [1, 2] и имеет большую практическую ценность (см., например, [3], [7], [8]). В настоящее время его реализации нет в стандартных библиотеках нескольких широко распространенных систем компьютерной алгебры, таких как Maple 9. Также актуальной является задача описания выпуклой оболочки конечного набора точек n–мерного пространства. В системе Maple 9 такая задача реализована лишь для двумерного случая.

1 Алгоритм Моцкина–Бургера

В дальнейшем изложении множеством констант является поле $\mathbb{Q}$. С теоретической точки зрения константы могут быть элементами любого упорядоченного поля (см., например, [1]). Для реализаций алгоритма уже необходимо упорядоченное поле с конструктивной арифметикой. В настоящее время алгоритм реализован лишь для случая поля рациональных чисел.

Дана система однородных линейных неравенств

$$l_j(x) = a_{j1}x_1 + \ldots + a_{jn}x_n \leqslant 0 \quad (j = 1, \ldots, m) \tag{1}$$

с коэффициентами $a_{ji} \in \mathbb{Q}$. Рангом r этой системы назовем ранг матрицы коэффициентов (a_{ji}).

Каждое из неравенств (1) определяет *полупространство* своих *решений* в $\mathbb{Q}^n$. Пересечение этих полупространств образует *выпуклый многогранный конус* (обозначим его C): если $X_1, \ldots, X_p$ являются решениями системы (1), то и любая их *коническая комбинация*

$$k_1X_1 + \ldots + k_pX_p, \qquad k_i \geqslant 0, \tag{2}$$

является решением системы (1). Назовем конечное число векторов $X_1, \ldots, X_p \in \mathbb{Q}^n$ *системой образующих* конуса C, если любое решение системы (1) представляется в виде (2) и ни один из векторов X_i не выражается в виде (2) через другие. Назовем множество решений P уравнения

$$l(x) = a_1x_1 + \ldots + a_nx_n = 0 \tag{3}$$

опорной гиперплоскостью к конусу C, если конус C лежит в полупространстве решений одного из неравенств $l(x) \leqslant 0$, $-l(x) \leqslant 0$. Множество $P \cap C$, где P — опорная гиперплоскость, назовем *гранью* конуса C. Грань конуса также является конусом; конус C является своей гранью. Пересечение конусов снова есть конус. *Размерностью конуса* d назовем размерность минимального линейного пространства, содержащего конус.

Добавим в систему (1) неравенство

$$l(x) = a_1x_1 + \ldots + a_nx_n \leqslant 0. \tag{4}$$

Вместе неравенства (1) и (4) определяют новый конус $C^\star$. Задача вычисления образующих конуса $C^\star$ по образующим конуса C решается алгоритмом Моцкина–Бургера, который заключается в следующем.

Конус $C^\star$ порождают образующие конуса C, лежащие в полупространстве решений неравенства $l(X) < 0$, и образующие пересечения конуса C и гиперплоскости P. Таким образом, те образующие X_i конуса C, для которых выполнено $l(X_i) \leqslant 0$, также являются образующими конуса $C^\star$, однако в общем случае они не порождают весь конус $C^\star$.

Переобозначим образующие $X_1, \ldots, X_p$ конуса C в соответствии со значением $l(x)$ на этих векторах: векторы, для которых $l(x) < 0$, $l(x) = 0$ и $l(x) > 0$, обозначим $X_1^-, \ldots, X_l^-$, $X_1^0, \ldots, X_s^0$ и $X_1^+, \ldots, X_k^+$ соответственно. Здесь $p = l + s + k$.

Для определения образующих пересечения конуса C и гиперплоскости P составим конические комбинации всех пар векторов X_α^+, X_β^-:

$$aX_\alpha^+ + bX_\beta^-, \quad a, b \geqslant 0. \tag{5}$$

Пересечение конуса таких комбинаций с гиперплоскостью P есть луч, образующий вектор которого

$$X_{\alpha\beta}^0 = -l(X_\beta^-)X_\alpha^+ + l(X_\alpha^+)X_\beta^-. \tag{6}$$

Среди векторов $X_{\alpha\beta}^0$ могут найтись "лишние", выражающиеся через другие векторы того же вида. Для исключения "лишних" векторов необходимо накладывать на пары X_α^+, X_β^- дополнительные ограничения. Система образующих конуса $C^\star$ содержится в множестве векторов $X_1^0, \ldots, X_s^0$, $X_1^-, \ldots, X_k^-$ и всех $X_{\alpha\beta}^0$.

В общем случае конус C есть сумма Минковского линейного подпространства T и заостренного конуса S, т.е. конуса, не содержащего ненулевого линейного подпространства. Обозначим базис максимального линейного подпространства T конуса C за U, множество образующих S — за V. Таким образом, $-U \cup U \cup V$ — система образующих конуса C. Элементы множества U будем обозначать за $u_1, \ldots, u_t$, элементы V — $v_1, \ldots, v_h$. Для конуса $C^\star$ соответствующие множества и элементы будем обозначать $T^\star$, $S^\star$, $U^\star$, $V^\star$.

Таким образом, записывая образующий элемент конуса C с верхним индексом $^+$, $^-$, ^{0}, мы описываем знак нового неравенства системы на этом элементе. Меняя букву X на буквы u или v, мы описываем знаки исходной системы неравенств на этом элементе. И, наконец, верхний индекс * означает принадлежность элемента к множеству образующих конуса $C^\star$.

Гиперплоскость P может содержать максимальное линейное подпространство T конуса C целиком, а может пересекать его. В первом случае для всех векторов $X \in U$ выполняется $l(X) = 0$, тем самым $U^\star = U$. Во втором случае пересечение $T^\star = T \cap P$ имеет размерность $t - 1$. Возьмем вектор из U, удовлетворяющий новому неравенству, например X_1^-. Тогда базис $T^\star$ составят $t - 1$ векторов вида

$$u_j^\star = -l(X_1^-)u_j + l(u_j)X_1^-. \tag{7}$$

Пересечение заостренного конуса S и гиперплоскости P в общем случае многомерно. При этом конус комбинаций (5) может пересекать гиперплоскость P по внутренности конуса C. Для исключения такой ситуации при пересчете образующих заостренного конуса S будем брать только пары $X_\alpha^+, X_\beta^- \in V$, лежащие на гранях размерности 2 конуса S, чтобы конус комбинаций (5) совпадал с гранью конуса S.

Чтобы узнать, принадлежит ли пара векторов $X_\alpha^+, X_\beta^- \in V$ грани размерности 2 конуса S, достаточно проверить, что все неравенства $l_j(x) \leqslant 0$ системы (1), для которых $l_j(X_\alpha^+) = l_j(X_\beta^-) = 0$, не обращаются в ноль каким-либо третьим элементом V (см. [1, с. 226]). Векторы X_α^+, X_β^-, для которых нет хотя бы одного неравенства системы (1) со свойством $l_j(X_\alpha^+) = l_j(X_\beta^-) = 0$, не лежат на грани размерности 2.

С помощью описанного алгоритма можно вычислить систему образующих самого конуса C, применив алгоритм сначала к неравенству $0 \leqslant 0$, а затем последовательно присоединяя к системе неравенства $l_j(x) \leqslant 0$ и применяя к получающимся системам неравенств описаный алгоритм.

Мы приходим к следующему алгоритму:

Вход: $L := \{l_1(x), \ldots, l_m(x)\}$ — множество левых частей неравенств системы (1).

Выход: $U = \{u_1, \ldots, u_t\}$ — базис линейного подпространства конуса решений системы (1), $V = \{v_1, \ldots, v_s\}$ — образующие заостренного конуса решений системы неравенств (1).

1. $U_{current} := \{(1, \ldots, 0), \ldots, (0, \ldots, 1)\}$; $V_{current} := \emptyset$; $L_{current} := \{0 \leqslant 0\}$; $i := 1$;
2. $l := l_i(x)$;

3. **if** $\exists u \in U_{current}$: $l(u) \neq 0$, **then**
$U_{current} := \{u_i l(u) - u l(u_i), \quad u_i \in U_{current}\}$;
$n := -(l(u)/|l(u)|)u$, $V_{current} := \{n, -v_i l(n) + n l(v_i), \quad v_i \in V_{current}\}$;
4. **else**
$V^1_{current} := \{v \in V_{current} |\ l(v) \leqslant 0\}$; $V^2_{current} := \emptyset$;
for $\forall (v_k, v_s) \in V^2_{current}\ :\ l(v_k) < 0,\ l(v_s) > 0$ **do**
$L^\star := \{l_j(x) \in L_{current} |\ l_j(v_k) = l_j(v_s) = 0\}$;
if $L^\star \neq \emptyset$ **then**
for $\forall\ v \in V_{current} \setminus \{v_k, v_s\}$
for $\forall\ l_j(x) \in L^\star$ **do**
if $l_j(v) \neq 0$ **then**
$V^2_{current} := V^2_{current} \cup \{-v_k l(v_s) + v_s l(v_k)\}$;
end if;
end do;
end do;
end if;
end do;
end if;
5. $V_{current} := V^1_{current} \cup V^2_{current}$;
6. $L_{current} := L_{current} \cup \{l_i(x)\}$; $L := L \setminus \{l_i(x)\}$; i:=i+1;
7. **if** $L = \emptyset$ **then**
goto 9;
end if;
8. goto 2;
9. $U := U_{current}$; $V := V_{current}$.

Перед применением данного алгоритма систему неравенств можно упростить (см. [9]).

2 Алгоритм вычисления выпуклой оболочки точек n-мерного пространства

Коническую комбинацию (2) конечного числа точек $S = \{Y_1, \ldots, Y_p\} \in V \subset \mathbb{Q}^n$ назовем *выпуклой*, если $\sum_{i=1}^{p} k_i = 1$. Совокупность всех выпуклых комбинаций точек $Y_1, \ldots, Y_p$ назовем *выпуклой оболочкой* этих точек, или *выпуклым многогранником* M, порожденным этими точками. Определим *опорную гиперплоскость* к M и *грань* многогранника M аналогично предыдущему пункту. Каждая грань M также является многогранником и, следовательно, выпуклой оболочкой своих вершин; грани образуют решетку по включению. Всякая грань может быть задана пересечением конечного числа гиперплоскостей. Назовем вектор, составленный из коэффициентов a_i в (3), *нормалью* к гиперплоскости (3). Размерностью d многогранника M также назовем размерность минимального линейного пространства, содержащего M. Будем говорить, что точка Y — внутренняя для многогранника M, если Y есть коническая комбинация порождающих M точек и не лежит ни на одной его собственной грани. Линейную оболочку точек $\{Y_1, \ldots, Y_p\}$ обозначим $L(Y_1, \ldots, Y_p)$.

Пусть $S = \{Y_1, \ldots, Y_{m+1}\}$ — конечное множество точек в $\mathbb{Q}^n$. Составим систему (1), определив

$$(a_{j1}, \ldots, a_{jn}) = Y_j - Y_{m+1}, \quad j = 1, \ldots, m. \tag{8}$$

Множество решений $N(Y_{m+1}, S)$ так определенной системы (1) назовем *нормальным конусом* точки Y_{m+1} относительно S. Пусть Y_{m+1} и Y_{m+2} — две точки множества $S = \{Y_1, \ldots, Y_{m+1}, Y_{m+2}\}$, тогда $N(Y_{m+1} \cup Y_{m+2}, S) = N(Y_{m+1}, S) \cap N(Y_{m+2}, S)$ назовем нормальным конусом множества $\{Y_{m+1} \cup Y_{m+2}\}$ относительно S. Вообще, нормальный конус любого подмножества $S' \subset S$ может быть определен пересечением $N(S', S) = \bigcap_{Y \in S'} N(Y, S)$ (см., например, [7]). Образующие нормальных конусов можно вычислить, используя описанный алгоритм Моцкина–Бургера.

Можно показать, что нормальный конус внутренней точки многогранника относительно множества порождающих его точек нулевой.

Обозначим нормальный конус $N(S,S)$ всего многогранника M, т.е. ортогональное дополнение к минимальному линейному пространству, содержащему M, за $\mathcal{N}$. Тогда $N(S',S)$ есть сумма Минковского $\mathcal{N}$ и $M(S',S)$ — проекции нормального конуса $N(S',S)$ на минимальное линейное пространство, содержащее M. Тогда $\mathcal{N}$ будет максимальным линейным подпространством конуса $N(S',S)$, а $M(S',S)$ — заостренным конусом. Множество образующих $M(S',S)$ обозначим за $V(M(S',S))$.

Пусть P — грань размерности s многогранника M. Вычислим ее нормальный конус $M(P,S) = \{k_1X_1 + \ldots + k_{n-s}X_{n-s}\}$. Каждый из X_i есть нормаль к некоторой гиперплоскости, определяющей грань P. Очевидно, что точка Y, лежащая на грани P, лежит также на каждой из этих гиперплоскостей. Значит, $\{k_1X_1 + \ldots + k_{n-s}X_{n-s}\} \subset M(Y,S)$. Также очевидно, что для точки Y, не лежащей на грани P, это не верно.

Для точки Y', лежащей на грани $P' \subset P$ размерности $s' < s$, верно $\{k_1X_1 + \ldots + k_{n-s}X_{n-s}\} \subset M(Y,S) \subset M(Y',S)$. Однако, если известно, что линейная оболочка точек $Y_1,\ldots,Y_m$ имеет размерность s, то размерность нормального конуса $M(\{Y_1,\ldots,Y_m\},S)$ в точности равна $n-s$ и $M(\{Y_1,\ldots,Y_m\},S) = \{k_1X_1 + \ldots + k_{n-s}X_{n-s}\}$. Таким образом, верна

Лемма 1. *Точки $Y_1,\ldots,Y_m$ лежат на грани P размерности s многогранника M с нормальным конусом $\{k_1X_1 + \ldots + k_{n-s}X_{n-s}\}$, тогда и только тогда, когда $\bigcap_{i=1}^m M(Y_i,S) = \{k_1X_1 + \ldots + k_{n-s}X_{n-s}\}$* и $\dim L(Y_1,\ldots,Y_m) = s$.

Приходим к следующему алгоритму:

Вход: $S := \{Y_1,\ldots,Y_m\}$ — множество точек в $\mathbb{Q}^n$.

Выход: S_j^i — список точек, попавших на j-ю грань размерности i многогранника M, порожденного точками S; N_j^i — список образующих нормальных конусов j-х граней размерности i.

1. Вычислим $N(S,S)$ и все $N(Y_i,S)$, используя алгоритм Моцкина–Бургера;
2. $\forall i \quad M(Y_i,S) := N(Y_i,S) \cap N(S,S)$;
3. $X := \bigcup_{Y_i\in S} V\,(M(Y_i,S))$ — общий список образующих векторов нормальных конусов всех точек Y_i;
4. **for** m **from** 1 **to** d **do**
   a. выберем в X j-ю m-ку векторов $(X_1,\ldots,X_m)$;
   b. выберем в S все точки $Y_1,\ldots,Y_s$, для которых $\{X_1,\ldots,X_m\} \subset V(M(Y_i,S))$;
   c. **if** таких точек $Y_1,\ldots,Y_s$ нет или $\dim L(Y_1,\ldots,Y_s) + \dim L(X_1,\ldots,X_m) \neq n$ **then**
      $j := j+1$;
      goto 4.a;
   end if;
   d. $N_j^m := \{X_1,\ldots,X_m\}$;
   e. $S_j^m := \{Y_1,\ldots,X_s\}$;
   end do.

3 Использование вычислительных пакетов

Пакет `motzkin_burger` содержит три процедуры, предназначенные для вызова пользователем: `DualForms`, `Conehull` и `MB`. Первая по заданному списку P точек из $\mathbb{Q}^n$ и его подсписку возвращает список линейных неравенств, задающих нормальный конус подмножества:

```
> P:=[[2,0,0],[1,1,1],[0,2,0],[1,0,1],[0,1,1],[0,0,0]]:
> L:=DualForms(P,[[2,0,0]],x,3);
```

```
L:=[-2*x[1], -x[1]+x[3], -2*x[1]+x[2]+x[3], -2*x[1]+2*x[2], -x[1]+x[2]+x[3]]
```

Здесь x — переменная, относительно которой вычислены неравенства, 3 — размерность пространства, $[[2, 0, 0]]$ — подсписок P.

Вторая процедура, `Conehull`, вычисляет базис линейного подпространства и множество образующих заостренного конуса решений заданного списка неравенств. Последние два аргумента несут тот же смысл. Так же есть четвертый опциональный параметр, строка "as is". Если он указан, перед итерированием алгоритма Моцкина–Бургера не производится предварительное упрощение исходной системы.

```
> Conehull(L,x,3);

            [], [[1, 1, 0], [0, -1, 0], [0, 0, -1], [1, 0, 1]]
```

Процедура `MB` производит пересчет образующих конуса решений системы при добавлении одного неравенства; здесь ее описывать не будем.

Пакет `convexhull` также содержит три процедуры, предназначенные для вызова пользователем: `Convexhull`, `FullConvexhull` и `Show3d`. Первая по заданному списку P точек из $\mathbb{Q}^n$ строит список S точек, лежащих на гипергранях выпуклой оболочки точек списка P, и список N образующих нормальных конусов соответствующих гиперграней (последний аргумент 3 — размерность пространства):

```
> S,N:=Convexhull(P,3): S; N;

            S :=[[[2, 0, 0], [1, 1, 1], [0, 2, 0]],
               [[1, 1, 1], [1, 0, 1], [0, 1, 1]],
               [[2, 0, 0], [1, 0, 1], [0, 0, 0]],
               [[2, 0, 0], [0, 2, 0], [0, 0, 0]],
               [[0, 2, 0], [0, 1, 1], [0, 0, 0]],
               [[2, 0, 0], [1, 1, 1], [1, 0, 1]],
               [[1, 1, 1], [0, 2, 0], [0, 1, 1]],
               [[1, 0, 1], [0, 1, 1], [0, 0, 0]]]

   N:=[[[1, 1, 0]], [[0, 0, 1]], [[0, -1, 0]], [[0, 0, -1]],
     [[-1, 0, 0]], [[1, 0, 1]], [[0, 1, 1]], [[-1, -1, 1]]]
```

Каждая из первых восьми строк — список точек, попавших на одну из восьми гиперграней, а последние две строки — список нормалей к соответствующим гиперграням. Так, на некоторой гиперграни лежат точки $(2, 0, 0)$, $(0, 2, 0)$, $(0, 0, 0)$.

Процедура `FullConvexhull` строит описание граней всех размерностей и нормальных конусов к ним. В обращении она аналогична `Convexhull`, но ответ более замысловат: `S[k]` — список списков точек, попавших на k–мерные грани, а `N[k]` — список нормалей к соответствующим k–мерным граням; например, список `S[k]` перечисляет все k-мерные грани, а его подсписок `S[k][1]` — точки, попавшие на первую k-мерную грань. В следующем примере вычисляется вся выпуклая оболочка, но выводится лишь список точек, попавших на первую из двумерных граней (совпадает с `S[1]` предыдущего примера):

```
> S,N:=FullConvexhull(P,3): S[2][1];

                [[2, 0, 0], [1, 1, 1], [0, 2, 0]]
```

Выпуклую оболочку списка точек P, вкладывающуюся в $\mathbb{Q}^3$ (сами точки списка P не обязательно должны иметь 3 или 2 координаты), можно увидеть, выполнив

```
> Show3d(P,3);
```

При написании кода вычислительных пакетов в полной мере использовалась "функциональная" парадигма программирования в среде Maple. Согласно последней, данные хранятся в виде упорядоченных списков; к переменным применяются встроенные процедуры Maple, которые над каждым элементом списка выполняют однотипные операции. Замеры производительности показывают, что использование "функциональной" парадигмы программирования дает существенный выигрыш во времени выполнения программы, в некоторых случаях до 10 раз.

Следующие таблицы показывают результаты замеров производительности процедур `Conehull` и `Convexhull` пакетов `motzkin_burger` и `convexhull` соответственно. Тестирование проводилось в среде

Maple 7 на компьютере на базе Duron 700Mhz, 256Mb RAM. Для оценки времени выполнения процедура `Convexhull` вызывалась 5 раз для случайного списка m точек n-мерного пространства. Указанное время t — среднее арифметическое времени выполнения `Convexhull`.

Аналогично, `Conehull` вызывалась 5 раз для списка случайных m неравенств ранга r на n переменных. Указанные значения t_1 и t_2 — времена выполнения процедуры с указанием параметра "as is"и без него, соответственно.

Производительность `Conehull`				
число переменных, n	число неравенств, m	ранг, r	`Conehull`, опция `"as is"`, t_1, sec	`Conehull`, t_2, sec
5	5	5	0.010	0.120
5	7	3	0.090	0.050
10	10	10	0.130	0.261
10	15	5	0.150	0.180
20	20	20	1.783	2.103
20	30	10	1.512	1.021
30	30	15	2.864	2.003
40	40	20	8.663	4.316
40	40	30	45.326	17.775
50	50	40	89.118	1816.392
50	50	45	73.766	1549.628

Производительность `Convexhull`		
число переменных, n	число неравенств, m	t, sec
3	10	2.890
3	30	28.661
3	50	94.157
4	10	5.332
4	30	214.073
10	10	19.408
20	20	301.493

Список литературы

[1] Черников С.Н. *Линейные неравенства* / С.Н.Черников. – М.: Наука, 1968.

[2] Солодовников А.С. *Системы линейных неравенств* / А.С.Солодовников. – М.: Наука, 1977.

[3] Осипов Н.Н. *Построение серий решетчатых кубатурных формул, точных на тригонометрических многочленах четырех переменных* / Н.Н.Осипов, А.В.Петров // Вычислительные технологии. – 2004. – Т. 9. Спец. вып. – С. 102-110.

[4] Емеличев В.А. *Многогранники, графы, оптимизация (комбинаторная теория многогранников)* / В.А.Емеличев, М.М.Ковалева, М.К.Кравцов. – М.: Наука, Глав. ред. физ.–мат. лит., 1981.

[5] Лейхтвейс К. *Выпуклые множества* / К.Лейхтвейс. – М.: Наука, 1985.

[6] Препарата Ф. *Вычислительная геометрия: введение* / Ф.Препарата, М.Шеймос. – М.: Мир, 1989.

[7] Брюно А.Д. *Степенная геометрия в алгебраических и дифференциальных уравнениях* / А.Д.Брюно. – М.: Наука, Физматлит. 1998.

[8] Soleev A. *Newton polyhedra and the reversible system of ordinary differential equations* / A.Soleev, I.Yarmukhamedov. – Proceedings of CASC 2000. – P. 335–343.

[9] Burovsky P.A. *Implementation of Motzkin-Burger algorithm in Maple* / P.A. Burovsky. arXiv:cs. CG/0501003.

MOTZKIN-BURGER ALGORITHM AND CONVEX HULLS COMPUTINGIN DIMENSION N

P.A.Burovsky

Implementation of Motzkin–Burger algorithm in Maple considered in this paper. Based on that one full convex hull description in dimension n algorithm implementation also presented.