СРАВНИТЕЛЬНОЕ ИЗУЧЕНИЕ ДВУХ БЫСТРЫХ АЛГОРИТМОВ ПРОЕКТИРОВАНИЯ ТОЧКИ НА СТАНДАРТНЫЙ СИМПЛЕКС*

Г. Ш. Тамасян
g.tamasyan@spbu.ru

15 мая 2014 г.

1°. В докладе [1] рассматривались два быстрых алгоритма проектирования точки на стандартный симплекс $\Lambda \subset \mathbb{R}^n$. Задача ортогонального проектирования точки $c = (c_1, \dots, c_n)$ на стандартный симплекс Λ, определяемый условиями

$$\sum_{i=1}^{n} x_i = 1; \quad x_i \geqslant 0,\ i \in 1:n,$$

ставится следующим образом:

$$Q(x) := \tfrac{1}{2} \sum_{i=1}^{n} (x_i - c_i)^2 \longrightarrow \min_{x \in \Lambda}. \tag{1}$$

Решение этой задачи существует и единственно. Обозначим его x^*.

2°. Начнём с описания алгоритма, идея которого предложена в [5]. Назовём его «векторным» алгоритмом.

Обозначим $N = 1 : n$.

Предварительный шаг. В качестве начального приближения возьмём вектор $x^{(0)}$ с компонентами

$$x_i^{(0)} = c_i + \lambda, \quad i \in N, \tag{2}$$

где

$$\lambda = \tfrac{1}{n}\Bigl(1 - \sum_{i \in N} c_i\Bigr). \tag{3}$$

*Семинар по конструктивному негладкому анализу и недифференцируемой оптимизации «CNSA & NDO»: `http://www.apmath.spbu.ru/cnsa/`

Общий шаг. Пусть имеется k-е приближение $x^{(k)}$. Если все компоненты вектора $x^{(k)}$ неотрицательны, то есть $x^{(k)} \geqslant \mathbb{O}$, то $x^{(k)}$ — искомая проекция точки c на стандартный симплекс Λ. Процесс заканчивается.

В противном случае, когда у вектора $x^{(k)}$ существует хотя бы одна отрицательная компонента, формируем индексное множество

$$I_k = \{i \in N \mid x_i^{(k)} \leqslant 0\} \tag{4}$$

и вычисляем

$$\lambda^{(k)} = \frac{1}{n_k}\Bigl(1 - \sum_{i \in N \setminus I_k} x_i^{(k)}\Bigr), \tag{5}$$

где $n_k = |N \setminus I_k|$ — мощность множества $N \setminus I_k$. В качестве очередного приближения берём вектор $x^{(k+1)}$ с компонентами

$$x_i^{(k+1)} = \begin{cases} 0 & \text{при } i \in I_k; \\ x_i^{(k)} + \lambda^{(k)} & \text{при } i \in N \setminus I_k. \end{cases} \tag{6}$$

После этого возвращаемся к общему шагу.

З а м е ч а н и е. Несложно убедиться в том, что $\lambda^{(k)}$ может быть вычислено по формуле

$$\lambda^{(k)} = \frac{1}{n_k} \sum_{i \in I_k} x_i^{(k)} \tag{7}$$

и, как следствие, принимает только отрицательное значение.

3°. В докладе [2] рассматривался другой алгоритм проектирования точки $c = (c_1, \ldots, c_n)$ на стандартный симплекс. Назовём его «скалярным» алгоритмом. Напомним его описание:

1) Меняем знаки у компонент c_j точки c и числа $\{-c_j\}$ упорядочиваем по неубыванию. Получаем последовательность $a_1 \leqslant \ldots \leqslant a_n$.

2) Проводим последовательные вычисления по рекуррентной формуле

$$\begin{gathered} \varphi_1 = 0, \\ \varphi_{k+1} = \varphi_k + k(a_{k+1} - a_k), \quad k = 1, \ldots, n-1, \end{gathered} \tag{8}$$

пока не встретим индекс k_0, на котором

$$\varphi_{k_0} < 1 \leqslant \varphi_{k_0+1}.$$

Если и $\varphi_n < 1$, то полагаем $k_0 = n$.

3) Вычисляем λ^* по формуле

$$\lambda^* = a_{k_0} + \tfrac{1}{k_0}(1 - \varphi_{k_0}). \tag{9}$$

Компоненты x_i^* проекции точки c на стандартный симплекс имеют вид

$$x_i^* = (\lambda^* + c_i)_+, \quad i \in 1 : n, \tag{10}$$

где $(u)_+ = \max\{0, u\}$.

4°. С точки зрения трудоёмкости, «скалярный» алгоритм выглядит предпочтительнее, поскольку в его основе лежит рекуррентное соотношение (8) для скалярных величин, в то время как в основе «векторного» алгоритма — рекуррентное соотношение (6) для векторных величин.

Максимальная трудоёмкость «векторного» алгоритма достигается тогда, когда у всех членов последовательности $x^{(0)}, x^{(1)}, \ldots$ имеется только по одной отрицательной компоненте. В этом случае $x^{(n-1)}$ совпадает с одним из ортов пространства $\mathbb{R}^n$.

В общем случае «векторный» алгоритм требует не более $n^2 + 2n - 1$ арифметических операций.

Максимальная трудоёмкость «скалярного» алгоритма достигается тогда, когда $\varphi_n < 1$. В общем случае «скалярный» алгоритм требует одну перестановку элементов массива длиной n и не более $4n - 2$ арифметических операций, где n — размерность пространства.

5°. В рассмотренных ранее примерах (см. доклад [1]) было замечено, что в случае, когда один из двух алгоритмов проектирования имеет максимальную трудоёмкость, у второго алгоритма трудоёмкость минимальна. Укажем конкретные области, точки из которых проектируются рассматриваемыми алгоритмами за минимальное и максимальное число шагов.

Введём функции

$$L_k(x) = nx_k - \sum_{i \in N} x_i + 1, \quad k \in N. \tag{11}$$

Нетрудно убедиться в том, что система неравенств $L_k(x) \geqslant 0$, $k \in N$ задаёт прямую призму, где одно из перпендикулярных сечений образует стандартный симплекс. В частности, при $n = 2$ данная система неравенств задаёт полосу (см. рис. 1). На рисунке стандартный симплекс выделен красным цветом.

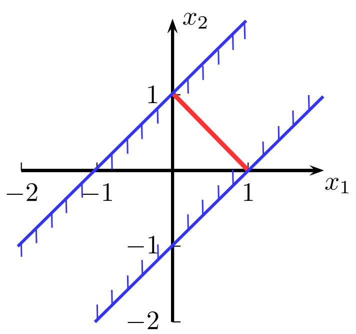

Рис. 1. Случай $n = 2$.

Пусть $\lambda = \frac{1}{n}\Big(1 - \sum\limits_{i \in N} c_i\Big)$.

ТЕОРЕМА 1. *Для того чтобы проекцией точки c на стандартный симплекс Λ была точка x^* с компонентами $x_k^* = c_k + \lambda$, $k \in N$, необходимо и достаточно, чтобы $L_k(c) \geqslant 0$ для всех $k \in N$.*

Доказательство. Необходимость. Пусть x^* искомая проекция точки c на Λ. Тогда $x_k^* = c_k + \lambda \geqslant 0$ для всех $k \in N$, т. е.

$$c_k + \frac{1}{n}\Big(1 - \sum_{i \in N} c_i\Big) \geqslant 0, \quad k \in N.$$

Если умножить на n обе части последних неравенств, то получим $L_k(c) = nc_k - \sum\limits_{i \in N} c_i + 1 \geqslant 0$ для всех $k \in N$.

Достаточность. Запишем критерий оптимальности для задачи (1) (см. [4, с. 91]):

$$\begin{gathered} x_i - c_i = \lambda + u_i, \quad i \in N; \\ u_i\, x_i = 0, \quad u_i \geqslant 0, \quad x_i \geqslant 0, \quad i \in N; \\ x_1 + \ldots + x_n = 1. \end{gathered} \tag{12}$$

Нетрудно проверить, что условия (12) выполняются при $\lambda = \frac{1}{n}\Big(1 - \sum\limits_{i \in N} c_i\Big)$, $u_i \equiv 0$, $x_i = c_i + \lambda$, $i \in N$. В частности, неравенство $L_i(c) = n(c_i + \lambda) \geqslant 0$, справедливо при всех $i \in N$ и обеспечивает выполнение условия $x_i \geqslant 0$, $i \in N$.

Теорема доказана. $\square$

Введём множества $M_k = \Big\{x \in \mathbb{R} \mid x_k - x_i \geqslant 1 \quad \forall i \in N \backslash \{k\}\Big\}$, $k = 1, 2$. На рис. 2 красным цветом выделен стандартный симплекс, а множества M_1 и M_2 образуют полуплоскости.

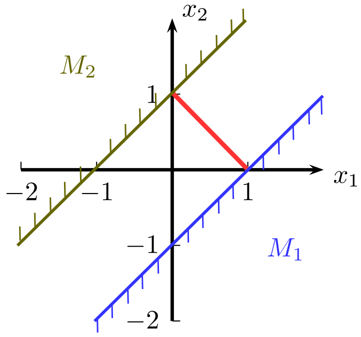

Рис. 2. Случай $n = 2$.

Обозначим через $\{e_k\}$, $k \in N$, — канонический базис в $\mathbb{R}^n$.

ТЕОРЕМА 2. *Для того чтобы проекцией точки c на стандартный симплекс Λ была точка $x^* = e_k$, необходимо и достаточно, чтобы $c \in M_k$.*

Доказательство. Достаточность. Пусть $c \in M_k$. Покажем, что $x^* = e_k$ — проекция точки c на Λ.

Запишем критерий оптимальности:

$$\begin{gathered} x_i - c_i = \lambda + u_i, \quad i \in N; \\ u_i\, x_i = 0, \quad u_i \geqslant 0, \quad x_i \geqslant 0, \quad i \in N; \\ x_1 + \ldots + x_n = 1. \end{gathered} \tag{13}$$

Нетрудно проверить, что условия (13) выполняются при $\lambda = 1 - c_k$, $x = e_k$, $u_k = 0$, $u_i = -\lambda - c_i$, $i \in N \backslash \{k\}$. Осталось показать, что $u_i = c_k - c_i - 1 \geqslant 0$ для всех $i \in N \backslash \{k\}$. Действительно, согласно условию $c \in M_k$, получим $c_k - c_i \geqslant 1$ для всех $i \in N \backslash \{k\}$.

Необходимость. Пусть x^* — искомая проекция точки c на Λ. Покажем, что $c \in M_k$.

Условия оптимальности (13) выполняются при $x^* = e_k$, $u_k = 0$, $\lambda = 1 - c_k$ и $u_i \geqslant 0$, $i \in N \backslash \{k\}$.

Рассмотрим равенства в первой строке условий (13) при $i \in N \backslash \{k\}$. Получим

$$u_i = \,-\lambda - c_i, \quad i \in N \backslash \{k\}.$$

На основании того, что $u_i \geqslant 0$ при всех $i \in N \backslash \{k\}$ и $\lambda = 1 - c_k$, приходим к соотношениям

$$u_i = c_k - c_i - 1 \geqslant 0, \quad i \in N \backslash \{k\}.$$

Выполнение этих неравенств и означает, что $c \in M_k$.

Теорема доказана. □

ТЕОРЕМА 3. *Максимальная трудоёмкость «скалярного» алгоритма достигается тогда и только тогда, когда $L_k(c) > 0$ при всех $k \in N$.*

Доказательство. Согласно «скалярному» алгоритму на первом шаге числа $\{-c_j\}$ упорядочиваем по неубыванию. Далее строим последовательность $a_1 \leqslant \ldots \leqslant a_n$ и вычисляем φ_k по рекуррентной формуле

$$\varphi_1 = 0, \quad \varphi_{k+1} = \varphi_k + k(a_{k+1} - a_k), \quad k = 1, \ldots, n-1.$$

Получаем

$$\begin{gathered} \varphi_1 = 0, \quad \varphi_2 = \,-a_1 + a_2, \quad \varphi_3 = \,-a_1 - a_2 + 2a_3, \\ \varphi_n = -a_1 - a_2 - \ldots - a_{n-1} + (n-1)a_n. \end{gathered}$$

Необходимость. Пусть $\varphi_n < 1$. Покажем, что $L_k(c) > 0$ при всех $k \in N$. Действительно, если вместо $\{a_i\}$ в неравенство $\varphi_n < 1$ подставить $\{-c_j\}$, получим

$$c_{k_1} + c_{k_2} + \ldots + c_{k_{n-1}} - (n-1)c_{k_n} < 1, \tag{14}$$

где $k_1, \ldots, k_n$ — некоторая перестановка чисел $1, \ldots, n$. Перепишем (14) в виде

$$nc_{k_n} - \sum_{j \in N} c_{k_j} + 1 > 0,$$

т. е. $L_k(c) > 0$ при всех $k \in N$.

Достаточность. Пусть $L_k(c) > 0$ при всех $k \in N$. Покажем, что $\varphi_n < 1$. Запишем систему неравенств $L_k(c) > 0$ при всех $k \in N$ в виде

$$nc_{k_n} - \sum_{j \in N} c_{k_j} + 1 > 0, \tag{15}$$

где $k_1, \ldots, k_n$ — некоторая перестановка чисел $1, \ldots, n$, а числа $\{-c_{k_j}\}$ упорядочены по неубыванию. Далее, согласно алгоритму строим последовательность $a_1 \leqslant \ldots \leqslant a_n$. Тогда неравенство (15) примет вид

$$-na_n + \sum_{j \in N} a_j + 1 > 0,$$

что эквивалентно тому, что $\varphi_n < 1$.

Теорема доказана. □

ТЕОРЕМА 4. *Минимальная трудоёмкость «скалярного» алгоритма достигается тогда и только тогда, когда $c \in M_k$ при некотором $k \in N$.*

Доказательство. Достаточность. Пусть $c \in M_k$, т. е.

$$-c_k \leqslant -c_i - 1 \quad \forall i \in N \backslash \{k\}. \tag{16}$$

В силу (16), на первом шаге «скалярного» алгоритма получим $a_1 = -c_k$. Далее вычислим $\varphi_1 = 0$ и $\varphi_2 = a_2 - a_1$, где a_2 может принять любое значение $\{-c_i\}$, $i \in N \backslash \{k\}$. Согласно условиям теоремы и (16) заключаем, что

$$\varphi_2 = a_2 - a_1 = c_k - c_i \geqslant 1, \quad i \in N \backslash \{k\}.$$

Таким образом, вычисления $\{\varphi_k\}$ закончились на первом шаге.

Н е о б х о д и м о с т ь. Пусть трудоёмкость «скалярного» алгоритма минимальна, т. е.

$$\varphi_2 = a_2 - a_1 \geqslant 1. \tag{17}$$

Здесь при некотором $k \in N$

$$a_1 := \min_{j \in N}\{-c_j\} = -c_k$$

и при некотором $i \in N \backslash \{k\}$

$$a_2 := -c_i.$$

Подставив $a_1 = -c_k$ и $a_2 = -c_i$ в (17), получим (16), т. е. $c \in M_k$.

Теорема доказана. □

ТЕОРЕМА 5. *Для того чтобы трудоёмкость «векторного» алгоритма была минимальной, необходимо и достаточно, чтобы $L_k(c) \geqslant 0$ при всех $k \in N$.*

Д о к а з а т е л ь с т в о немедленно следует из теоремы 1 и предварительного шага «векторного» алгоритма (см. (2), (3)). □

СЛЕДСТВИЕ 1. *Из теорем 1, 3 и 5 следует, что, если при проектировании точки c на стандартный симплекс Λ у «скалярного» алгоритма достигается максимальная трудоёмкость, то для «векторного» алгоритма трудоёмкость минимальна, а именно, «векторный» алгоритм закончится на предварительном шаге.*

Пусть $\ell_1, \ldots, \ell_n$ — некоторая перестановка чисел $\{1, \ldots, n\}$.

ТЕОРЕМА 6. *Трудоёмкость «векторного» алгоритма максимальна тогда и только тогда, когда найдется такая перестановка $\ell_1, \ldots, \ell_n$ при которой компоненты точки c удовлетворяют следующим неравенствам:*

$$\begin{cases} c_{\ell_{j-1}} < c_{\ell_j} + \left[(n-j)c_{\ell_j} - \displaystyle\sum_{k=j+1}^{n} c_{\ell_k} + 1\right], \quad j = 2, \ldots, n-1, \\ c_{\ell_{n-1}} < c_{\ell_n} - 1. \end{cases} \tag{18}$$

Доказательство теоремы опирается на некоторые вспомогательные утверждения.

Далее для простоты изложения будем считать, что $\ell_j = j,\ j \in N$. Предположим, что компоненты точки $c \in \mathbb{R}^n$ удовлетворяют условиям (18). Тогда неравенства (18) примут вид

$$\begin{cases} c_{n-1} < c_n - 1, \\ c_{n-2} < c_{n-1} + \Big[c_{n-1} - c_n + 1\Big], \\ c_{n-3} < c_{n-2} + \Big[2c_{n-2} - c_{n-1} - c_n + 1\Big], \\ \quad \vdots \\ c_{k-1} < c_k + \Big[(n-k)c_k - \sum\limits_{j=k+1}^{n} c_j + 1\Big], \quad k = 2, \ldots, n-1, \\ \quad \vdots \\ \ c_2 < c_3 + \Big[(n-3)c_3 - \sum\limits_{k=4}^{n} c_k + 1\Big], \\ \ c_1 < c_2 + \Big[(n-2)c_2 - \sum\limits_{k=3}^{n} c_k + 1\Big]. \end{cases} \tag{19}$$

ЛЕММА 1. *Если компоненты точки $c \in \mathbb{R}^n$ удовлетворяют условиям* (19)*, то*

$$c_1 < c_2 < \ldots < c_{n-1} < c_n - 1. \tag{20}$$

Значит, точка c принадлежит множеству $\operatorname{int} M_n$ (см. (16)).

Доказательство. Достаточно показать, что выражения в квадратных скобках в правых частях неравенств (19) отрицательные.

Действительно, выражение в квадратных скобках во втором неравенстве меньше нуля в силу первого неравенства. Тогда получим, что $c_{n-2} < c_{n-1}$, а следовательно $c_{n-2} < c_n - 1$. Отсюда имеем, что в третьем неравенстве выражение в квадратных скобках меньше нуля. Аналогично доказываются и оставшиеся неравенства.

Таким образом можно показать, что $c_n - c_k > 1$ для всех $k \in 1 : n-1$, откуда и следует, что точка c принадлежит множеству $\operatorname{int} M_n$.

Лемма доказана. $\square$

СЛЕДСТВИЕ 2. *Пусть W — множество точек c из $\mathbb{R}^n$ удовлетворяющих условиям* (18)*. Тогда $W \subset \operatorname{int} M_{\ell_n}$.*

ЛЕММА 2. *Если компоненты точки $c \in \mathbb{R}^n$ удовлетворяют условиям* (19), *то*

$$\begin{cases} L_1(c) = nc_1 - \sum\limits_{j\in N} c_j + 1 < 0, \\ L_2(c) = nc_2 - \sum\limits_{j\in N} c_j + 1 > 0, \\ \quad\vdots \\ L_{n-1}(c) = nc_{n-1} - \sum\limits_{j\in N} c_j + 1 > 0, \\ L_n(c) = nc_n - \sum\limits_{j\in N} c_j + 1 > n. \end{cases} \tag{21}$$

Доказательство. Справедливость первого и последнего неравенств следует из (20). Покажем, что $L_k(c) > 0$ для всех $k = 2, \ldots, n-1$. Перепишем выражение для $L_k(c)$ в следующем виде:

$$L_k(c) = nc_k - \sum_{j\in N} c_j + 1 =$$

$$= \left(c_k - c_{k-1} + \left[(n-k)c_k - \sum_{j=k+1}^{n} c_j + 1\right]\right) + \left(kc_k - \sum_{j=1}^{k} c_j\right) - c_k + c_{k-1} =$$

$$= \left(c_k - c_{k-1} + \left[(n-k)c_k - \sum_{j=k+1}^{n} c_j + 1\right]\right) + \sum_{j=1}^{k-2}(c_k - c_j).$$

Первая группа слагаемых положительна в силу k-го неравенства в (19), а положительность второго слагаемого следует из (20). □

ЛЕММА 3. *Если компоненты точки $c \in \mathbb{R}^n$ удовлетворяют условиям* (19), *то*

$$\sum_{j=1}^{k-1} c_j - (k-1)c_k + L_k(c) < 0 \quad \text{для всех } k = 2, \ldots, n-1, \tag{22}$$

$$\sum_{j=1}^{k-1} c_j - (k-1)c_\ell + L_\ell(c) > 0 \quad \text{при } k+1 \leqslant \ell \leqslant n-1, \tag{23}$$

$$\sum_{j=1}^{k-1} c_j - (k-1)c_n + L_n(c) > n-k+1. \tag{24}$$

Доказательство. Вначале рассмотрим неравенства (22). С учетом (11), преобразуем левую часть (22):

$$\sum_{j=1}^{k-1} c_j - (k-1)c_k + L_k(c) =$$

$$= \sum_{j=1}^{k-1} c_j - (k-1)c_k + nc_k - \sum_{j\in N} c_j + 1 = (n-k)c_k - \sum_{j=k+1}^{n} c_j + 1 =$$

$$= \sum_{j=k+1}^{n-1}(c_k - c_j) + (c_k - c_n + 1).$$

Отсюда, и в силу (20), следует справедливость неравенств (22).

Теперь докажем неравенства (23). Используя то, что для допустимых ℓ выполнено неравенство $c_\ell \geqslant c_{k+1}$ (см. (20)), преобразуем левую часть (23) к виду:

$$\sum_{j=1}^{k-1} c_j - (k-1)c_\ell + L_\ell(c) = c_\ell - c_k + \sum_{j=k+1}^{n} (c_\ell - c_j) + 1 \geqslant$$
$$\geqslant c_{k+1} - c_k + \sum_{j=k+1}^{n} (c_\ell - c_j) + 1 > 0.$$

Последнее полученное неравенство является следствием одного из неравенств (19).

Справедливость (24) следует из того, что $c_n - c_j > 1$ для всех $j \in N$. Действительно, рассмотрим и преобразуем левую часть (24):

$$\sum_{j=1}^{k-1} c_j - (k-1)c_n + L_n(c) = \sum_{j=1}^{k-1} c_j - (k-1)c_n + nc_n - \sum_{j\in N} c_j + 1 =$$
$$= \sum_{j=k}^{n-1} (c_n - c_j) + 1 > n - k + 1.$$

Лемма 3 доказана. □

Доказательство теоремы 6. Необходимость. Пусть трудоёмкость «векторного» алгоритма максимальная. Это означает, что индексные множества I_k (см. (4)) на каждом шаге пополнялись только одним элементом. Не умаляя общности, предположим, что $I_0 = \{1\}$, $I_k = I_{k-1} \bigcup \{k+1\}$, $k = 1, \ldots, n-2$. Нетрудно заметить, что в этом случае $\ell_n = n$. И вновь, для простоты изложения, будем считать, что $\ell_j = j$, $j \in N$. Покажем, что тогда точка c удовлетворяет неравенствам (19) и, согласно теореме 2, проекцией точки c на стандартный симплекс Λ будет точка $x^* = e_n$.

На предварительном шаге первого алгоритма находим $\lambda = \frac{1}{n}\left(1 - \sum\limits_{j\in N} c_j\right)$ и вектор $x^{(0)}$ с компонентами

$$x_i^{(0)} = c_i + \lambda = \tfrac{1}{n}\left(nc_i - \sum_{j\in N} c_j + 1\right), \quad i \in N.$$

По предположению на первом шаге имеем $I_0 = \{1\}$, тогда (см. (11))

$$\begin{cases} x_1^{(0)} = \frac{1}{n} L_1(c) < 0, \\ x_2^{(0)} = \frac{1}{n} L_2(c) > 0, \\ \quad\vdots \\ x_{n-1}^{(0)} = \frac{1}{n} L_{n-1}(c) > 0, \\ x_n^{(0)} = \frac{1}{n} L_n(c) > 0. \end{cases}$$

Заметим, что если в последнем неравенстве в (19) c_1 перенести в правую часть, то получим $L_2(c)$, которое положительно. Таким образом, справедливость последнего неравенства в (19) показана.

Далее находим (см. (5)–(7))

$$\lambda^{(0)} = \frac{1}{n-1} x_1^{(0)} = \frac{1}{n(n-1)} L_1(c).$$

Первое приближение принимает вид

$$x^{(1)} = \begin{pmatrix} 0 \\ x_2^{(0)} + \lambda^{(0)} \\ \vdots \\ x_n^{(0)} + \lambda^{(0)} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{(n-1)}\big(c_1 - c_2 + L_2(c)\big) \\ \vdots \\ \frac{1}{(n-1)}\big(c_1 - c_n + L_n(c)\big) \end{pmatrix}.$$

На втором шаге индексное множество I_1 состоит из двух элементов $\{1, 2\}$, то есть

$$\begin{cases} x_1^{(1)} = 0, \\ x_2^{(1)} = \frac{1}{(n-1)}\big(c_1 - c_2 + L_2(c)\big) < 0, \\ x_3^{(1)} = \frac{1}{(n-1)}\big(c_1 - c_3 + L_3(c)\big) > 0, \\ \quad\vdots \\ x_{n-1}^{(1)} = \frac{1}{(n-1)}\big(c_1 - c_{n-1} + L_{n-1}(c)\big) > 0, \\ x_n^{(1)} = \frac{1}{(n-1)}\big(c_1 - c_n + L_n(c)\big) > 0. \end{cases}$$

Теперь, если в предпоследнем неравенстве в (19) c_2 перенести в правую часть, то получим выражение $c_1 - c_3 + L_3(c)$, которое положительно, т. к. $x_3^{(1)} > 0$. Таким образом, справедливость предпоследнего неравенства в (19) также показана.

Осталось вычислить:

$$\lambda^{(1)} = \frac{1}{n-2} x_2^{(1)} = \frac{1}{(n-1)(n-2)} \left(c_1 - c_2 + L_2(c)\right).$$

Второе приближение примет вид

$$x^{(2)} = \begin{pmatrix} 0 \\ 0 \\ x_3^{(1)} + \lambda^{(1)} \\ \vdots \\ x_n^{(1)} + \lambda^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{(n-2)}\big(c_1 + c_2 - 2c_3 + L_3(c)\big) \\ \vdots \\ \frac{1}{(n-2)}\big(c_1 + c_2 - 2c_n + L_n(c)\big) \end{pmatrix}.$$

Наконец, на последнем шаге будем иметь $I_{n-2} = \{1, \ldots, n-1\}$. Тогда

$$\begin{cases} x_1^{(n-2)} = \ldots = x_{n-2}^{(n-2)} = 0, \\ x_{n-1}^{(n-2)} = \frac{1}{2}\left(\sum\limits_{i=1}^{n-2} c_i - (n-2)c_{n-1} + L_{n-1}(c)\right) < 0, \\ x_n^{(n-2)} = \frac{1}{2}\left(\sum\limits_{i=1}^{n-2} c_i - (n-2)c_n + L_n(c)\right) > 1. \end{cases} \tag{25}$$

Преобразовав последнее неравенство, получим $c_n - c_{n-1} - 1 > 0$. Таким образом, справедливость первого неравенства в (19) доказана.

Теперь вычислим

$$\lambda^{(n-2)} = \frac{1}{2}\left(\sum_{i=1}^{n-2} c_i - (n-2)c_{n-1} + L_{n-1}(c)\right).$$

Отсюда, и в силу замечания, что все $\lambda^{(k)}$ принимают только отрицательные значения, получим неравенство

$$\lambda^{(n-2)} = c_{n-1} + 1 - c_n < 0. \tag{26}$$

В итоге, последнее приближение примет вид

$$x^{(n-1)} = e_n = x^*,$$

т. к. $x_i^{(n-1)} = 0$ при $i \in 1 : n-1$ и $x_n^{(n-1)} := x_n^{(n-2)} + \lambda^{(n-2)} = 1$.

Аналогично можно рассмотреть иные закономерности образования индексных множеств I_k, $k = 1, \ldots, n-2$, из элементов $\{1, \ldots, n-1\}$ и получить следующие неравенства (см. (26))

$$c_i + 1 - c_n < 0, \quad i = 1, \ldots, n-1, \tag{27}$$

а это и означает, что точка c принадлежит множеству $\operatorname{int} M_n$.

Достаточность. Пусть компоненты точки c удовлетворяют неравенствам (19). По лемме 2 имеем $I_0 = \{1\}$. Следуя алгоритму, описанному выше, можно показать, что, в силу леммы 3, индексные множества I_k на каждом шаге пополняются только одним элементом. А именно,

$$I_k = I_{k-1} \bigcup \{k+1\}, \quad k = 1, \dots, n-2.$$

Таким образом, показана максимальная трудоёмкость «векторного» алгоритма.

Теорема доказана. $\square$

СЛЕДСТВИЕ 3. *Из теорем 2, 4, 6 и следствия 2 следует, что, если при проектировании точки c на стандартный симплекс Λ у «векторного» алгоритма достигается максимальная трудоёмкость, то для «скалярного» алгоритма трудоёмкость минимальна.*

ЛИТЕРАТУРА

1. Малоземов В. Н., Тамасян Г. Ш. *Ещё один быстрый алгоритм проектирование точки на стандартный симплекс* // Семинар «DHA & CAGD». Избранные доклады. 5 сентября 2013 г. (`http://dha.spb.ru/reps13.shtml#0905`)

2. Малоземов В. Н. *Проектирование точки на подпространство и на стандартный симплекс* // Семинар «DHA & CAGD». Избранные доклады. 28 февраля 2013 г. (`http://dha.spb.ru/reps13.shtml#0228`)

3. Малоземов В. Н., Певный А. Б. *Быстрый алгоритм проектирования точки на симплекс* // Вестник СПбГУ. Сер. 1. 1992. Вып. 1 (No 1). С. 112–113.

4. Гавурин М. К., Малоземов В. Н. Экстремальные задачи с линейными ограничениями. Л.: Изд-во ЛГУ, 1984. 176 с.

5. Michelot C. *A finite algorithm for finding the projection of a point onto the canonical simplex of* $\mathbb{R}^n$ // JOTA. 1986. Vol. 50. No 1. P. 195–200.

6. Causa A., Raciti F. *A purely geometric approach to the problem of computing the projection of a point on a simplex* //JOTA. 2013. Vol. 156. No 2. P. 524–528.