ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель–июнь 2018. Том 25, № 2. С. 82–100

УДК 519.8 DOI: 10.17377/daio.2018.25.571

ОЦЕНКА ТРУДОЁМКОСТИ АЛГОРИТМА ПО ПОИСКУ НУЛЯ ОДНОЙ ВЫПУКЛОЙ КУСОЧНО-ЛИНЕЙНОЙ ФУНКЦИИ

Е. В. Просолупов[a], *Г. Ш. Тамасян*[b]

Санкт-Петербургский гос. университет,
Университетский пр., 35, 198504 Санкт-Петербург, Россия
E-mail: [a]e.prosolupov@spbu.ru, [b]g.tamasyan@spbu.ru

Аннотация. Известно, что задача ортогонального проецирования точки на стандартный симплекс сводится к решению скалярного уравнения. В работе анализируется трудоёмкость алгоритма поиска нуля выпуклой кусочно-линейной функции, предложенного в [30]. Проведён анализ лучших и худших случаев входных данных для алгоритма. Для этого изучены наибольшее и наименьшее числа итераций алгоритма как функции от размера входных данных. Показано, что в случае равенства элементов входного множества алгоритм совершает наименьшее число итераций. В случае же различающихся элементов входного множества количество итераций максимально и очень слабо зависит от конкретных значений элементов множества. Приведены результаты вычислительных экспериментов со случайными входными данными высокой размерности. Табл. 2, ил. 2, библиогр. 34.

Ключевые слова: стандартный симплекс, ортогональное проецирование точки, нули функции.

Введение

Во многих прикладных задачах (астрономии, распознавания образов, математической диагностики, теории управления), а также в методах математического программирования (условной и безусловной оптимизации) требуется найти проекцию точки на множество [2–5, 8, 11–16, 18, 19, 22–25, 27, 32, 33]. Очень часто используется евклидова норма, однако встречаются и другие нормы [6, 7] (в приложении к экономическим задачам).

Известно, что задача ортогонального проецирования точки на стандартный симплекс ставится как задача квадратичного программирования [8, 32]. В [28] было впервые представлено своде́ние данной задачи

© 2018 Е. В. Просолупов, Г. Ш. Тамасян

к решению скалярного уравнения. Позднее в [20] был предложен алгоритм для решения более общей задачи (квадратичного программирования), имеющий линейный порядок сложности. Отметим также работы [9, 26, 29], в которых решение данного скалярного уравнения находят другими методами.

На ином принципе устроен алгоритм поиска проекции точки на стандартный симплекс, предложенный в [31]. Он имеет наглядную геометрическую интерпретацию, что отмечено в [10, 21]. Вопросу трудоёмкости алгоритма из [31] посвящены работы [10, 17, 34].

В статье анализируется трудоёмкость алгоритма по поиску нуля скалярного уравнения, предложенного в [30]. На идейном уровне он близок к работе [20]. Проведён анализ лучших и худших случаев входных данных для алгоритма. Для этого изучены наибольшее и наименьшее числа итераций алгоритма как функции от размера входных данных. Показано, что в случае равенства элементов входного множества алгоритм совершает наименьшее число итераций. В случае же различающихся элементов входного множества количество итераций максимально и очень слабо зависит от конкретных значений элементов множества. Приведены результаты вычислительных экспериментов со случайными входными данными высокой размерности.

Статья организована следующим образом. В разд. 1 и 2 дана формальная постановка задачи и описание алгоритма. Разд. 3 посвящён анализу трудоёмкости алгоритма, а разд. 4 содержит результаты численных экспериментов.

1. Постановка задачи

Пусть задано множество вещественных чисел $\{c_1, \ldots, c_n\}$ и положительная константа β. Обозначим $N = 1 : n$. Рассмотрим функцию φ от скалярной переменной t:

$$\varphi(t) = \sum_{j \in N} \max\{c_j - t, 0\}. \tag{1}$$

Задача. *Решить уравнение*

$$\varphi(t) = \beta. \tag{2}$$

Нетрудно заметить, что $\varphi(t)$ — непрерывная кусочно-линейная выпуклая функция на всей вещественной оси (см. рис. 1).

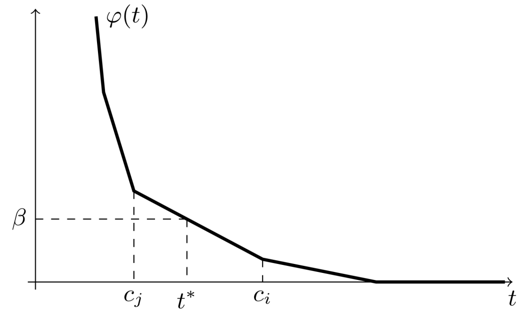

Рис. 1. График функции $\varphi(t)$

Более того, при $t \leqslant \max\limits_{j\in N} c_j$ функция $\varphi(t)$ монотонно убывает и равна нулю при $t \geqslant \max\limits_{j\in N} c_j$. Таким образом, уравнение (2) имеет решение, и это решение единственно. Обозначим его через t^*.

При $\beta = 1$ задача (2) возникает в процессе поиска ортогональной проекции точки на стандартный симплекс (см. [9, 10, 20, 21, 28–30, 32]).

Подробно изучим алгоритм решения уравнения (2), описанный в [30].

2. Алгоритм Maculan — Galdino de Paula (Maculan)

Далее нам потребуется понятие медианы множества.

Определение 1. Пусть $S = \{s_1, \ldots, s_\ell\}$ — множество вещественных чисел. Обозначим упорядоченный по неубыванию набор элементов множества S через $(a_1, \ldots, a_\ell)$. Под *медианой M множества S* понимается величина $a_m \in S$, которая стоит на позиции $m = \left\lfloor \frac{\ell+1}{2} \right\rfloor$, т. е.

$$M = a_m,$$

где $\lfloor z \rfloor$ — целая часть числа z.

Переходим к описанию алгоритма.

Инициализация.

$$S := \{c_1, \ldots, c_n\}, \quad J := N, \quad v := 0, \quad p := 0, \quad q := 0.$$

Введём функцию

$$\Phi(t, x, L, v, p, q) := \sum_{j\in L} (x_j - t) + v + p(q - t), \tag{3}$$

где $x = (x_1, \ldots, x_n)$, $L \subseteq N$.

Шаг 1. Имеем множество $S = \{c_j \mid j \in J\}$. Найдём медиану M множества S и построим следующие индексные множества:

$$L_= := \{j \in J \mid c_j = M\},$$
$$L_> := \{j \in J \mid c_j > M\},$$
$$L_< := \{j \in J \mid c_j < M\}.$$

Пусть $m \in L_=$, т. е.

$$c_m = M. \tag{4}$$

Вычислим

$$z = \Phi(M, c, L_>, v, p, q). \tag{5}$$

Если $z \geqslant \beta$, то переходим к шагу 2, иначе к шагу 3.

Шаг 2. Положим

$$J := L_> \cup \{m\}, \quad S := \{c_j \mid j \in J\},$$

где m — индекс, найденный на шаге 1 (см. (4)).

Если $|J| \geqslant 3$, то переходим к шагу 1, иначе при $v = z$ и $q = M$ — к шагу 4. Здесь $|J|$ — мощность индексного множества J.

Шаг 3. Положим

$$J := L_< \cup \{m\}, \quad v := z, \quad q := M, \quad p := p + |L_=| - 1 + |L_>|.$$

Если $|J| \geqslant 2$, то переходим к шагу 1, иначе к шагу 4.

Шаг 4. Решение уравнения (2) находим по формуле

$$t^* := q - \frac{\beta - v}{1 + p}. \tag{6}$$

Описание алгоритма завершено.

Замечание 1. Если индексное множество $L_=$ состоит более чем из одного элемента, то в качестве c_m в формуле (4) можно выбрать любой элемент множества S, равный медиане M.

Замечание 2. Величина z, вычисляемая на шаге 1 (см. (5)), является значением функции $\varphi(t)$ (см. (1)) в точке t, равной медиане M, т. е.

$$\Phi(M, c, L_>, v, p, q) = \varphi(M).$$

Замечание 3. Мощность индексного множества J от итерации к итерации строго убывает (практически в два раза). Это гарантирует конечность процесса.

На рис. 2 приведена схема работы алгоритма Maculan. Каждая итерация алгоритма, кроме последней, состоит из выполнения последовательности шаг $1 +$ шаг 2 в одном случае или шаг $1 +$ шаг 3 с последующим возвратом к шагу 1 — в другом. Переход от шага 1 к шагу 2 осуществляется, если значение функции φ в текущей медиане c_m больше или равно β и требуется продолжить поиск в правой половине списка c_j ($c_j \geqslant c_m$). Переход от шага 1 к шагу 3 осуществляется в противном случае. Тогда поиск продолжается в левой части набора ($c_j \leqslant c_m$). Если будет выполнен переход к шагу 4, то после его выполнения алгоритм завершит свою работу.

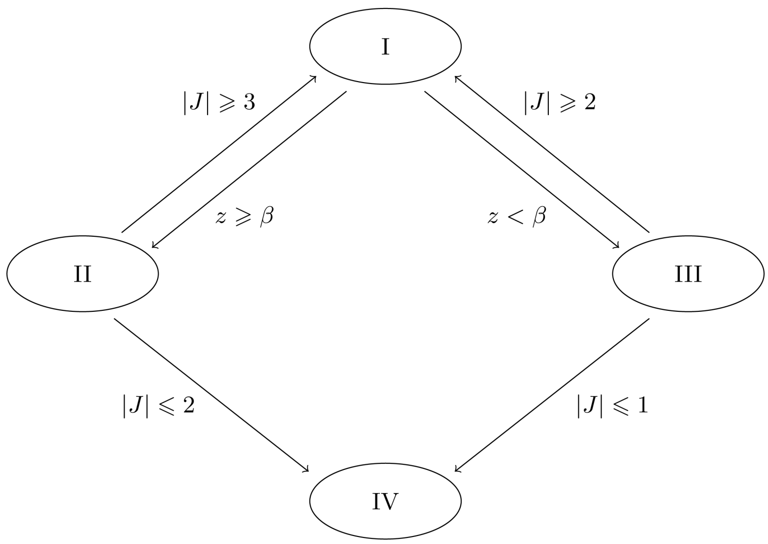

Рис. 2. Схема алгоритма:
I — шаг $1, \ldots,$ IV — шаг 4

Отметим, что приведённый алгоритм имеет линейный порядок сложности. Известно, что сложность поиска медианы в числовом множестве из n элементов составляет $O(n)$ [1]. Вычисление функции $\Phi(t, x, L, v, p, q)$ имеет трудоёмкость $O(|L|)$. Таким образом, все действия шага 1 имеют порядок сложности $O(|J|)$. Если обозначить через n_i мощность множества J на i-й итерации, то согласно описанию шага 2 и шага 3 получаем

$$n_{i+1} \leqslant \frac{n_i + 1}{2} = \frac{n_i}{2} + \frac{1}{2}.$$

Тем самым количество действий на всех итерациях алгоритма имеет порядок

$$O(n) + O\left(\frac{n}{2} + \frac{1}{2}\right) + O\left(\frac{n}{4} + \frac{3}{4}\right) + O\left(\frac{n}{8} + \frac{7}{8}\right) + \cdots + O(2) = O(n).$$

3. О лучших и худших случаях

Как уже упоминалось выше, любая итерация алгоритма Maculan состоит из шага 1, за которым следует шаг 2 или шаг 3 (см. схему алгоритма на рис. 2). Оценим, при каких входных данных (числовом множестве $\{c_1, \ldots, c_n\}$ и положительной константе β) алгоритм выполняет максимальное, а при каких — минимальное число итераций.

Легко показать, что для любого $n \geqslant 1$ задача (2) решается за одну итерацию в случае равенства всех элементов множества $\{c_1, \ldots, c_n\}$. Действительно, при первом прохождении шага 1 будут образованы множества

$$L_{=} := 1 : n, \quad L_{>} := \varnothing, \quad L_{<} := \varnothing,$$

т. е. независимо от выбора следующего шага множество J при выходе из шага 2 или шага 3 будет содержать только один элемент и алгоритм перейдёт к шагу 4.

Таким образом, естественно считать случай равных элементов множества $\{c_1, \ldots, c_n\}$ лучшим случаем входных данных для алгоритма Maculan.

3.1. Наибольший и наименьший размер множества для данного числа итераций. Во всех рассуждениях далее будем предполагать, что $c_i \neq c_j$ для всех $i \neq j$. В рамках сделанного предположения рассмотрим минимальное и максимальное значения параметра n как функции от количества итераций.

Обозначим через $d(k)$ и $D(k)$ соответственно минимальное и максимальное значение n, для которого возможны такие начальные условия, чтобы алгоритм Maculan решал задачу за k итераций.

Пусть алгоритм Maculan решает задачу (2) за $\widehat{k}$ итераций. Тогда справедливы следующие неравенства:

$$d(\widehat{k}) \leqslant n \leqslant D(\widehat{k}). \tag{7}$$

В рамках сделанного предположения о неравенстве элементов входного множества $\{c_1, \ldots, c_n\}$ можно найти явные формулы для величин $d(k)$ и $D(k)$.

Теорема 1. *Выполняется равенство $d(1) = 1$, и для всех $k \geqslant 2$ справедлива формула*

$$d(k) = 2^{k-2} + 2. \tag{8}$$

Доказательство. Очевидно, что $d(1) = 1$. Можно убедиться, что $d(2) = 3$. Это количество шагов достигается, если во время первой итерации $z < \beta$. Справедливо рекуррентное соотношение

$$d(k+1) = 2 \cdot (d(k) - 1), \quad k \geqslant 2. \tag{9}$$

Чтобы доказать это, изучим изменение мощности n входного множества $\{c_1, \ldots, c_n\}$ от итерации к итерации.

Рассмотрим входное множество $\{c_1, \ldots, c_n\}$, для которого потребуется $k+1$ итераций алгоритма Maculan. После первой итерации алгоритма будет получено такое множество $J = \{j_1, \ldots, j_{n_1}\}$, что для множества $\{c_{j_1}, \ldots, c_{j_{n_1}}\}$ требуется k итераций алгоритма.

Если первая итерация состояла из последовательности шаг 1 + шаг 2 (т. е. на шаге 1 было верно неравенство $z \geqslant \beta$), то $n = 2 \cdot (n_1 - 1)$ в случае чётного n или $n = 2 \cdot n_1 - 1$ в случае нечётного.

Если первая итерация была шаг 1 + шаг 3 (при $z < \beta$), то $n = 2 \cdot n_1$ или $n = 2 \cdot n_1 - 1$ соответственно.

Таким образом, наименьшее возможное значение мощности входного множества $d(k+1)$, $k \geqslant 2$, из которого после первой итерации можно получить множество мощности $d(k)$, это $2 \cdot (d(k) - 1)$ (при условии, что на первой итерации $z \geqslant \beta$). Формула (9) доказана.

Приведём рекуррентное соотношение (9) к замкнутой форме. Рассмотрим функцию $Y(k) = d(k) - 2$. При $k \geqslant 2$ имеем

$$\begin{aligned} Y(k+1) + 2 = d(k+1) = 2 \cdot (d(k) - 1) \\ = 2 \cdot (Y(k) + 2 - 1) = 2 \cdot Y(k) + 2. \end{aligned}$$

Таким образом,

$$Y(k+1) = 2 \cdot Y(k), \quad k \geqslant 2,$$

и $Y(2) = d(2) - 2 = 1$. Следовательно,

$$\begin{gathered} Y(k) = 2^{k-2}, \quad k \geqslant 2, \\ d(k) = Y(k) + 2 = 2^{k-2} + 2, \quad k \geqslant 2. \end{gathered}$$

Теорема 1 доказана.

Теорема 2. *Для всех $k \geqslant 1$ справедлива формула*

$$D(k) = 3 \cdot 2^{k-1}. \tag{10}$$

Доказательство. Несложно убедиться, что $D(1) = 3$. Действительно, всего одна итерация в случае $n = 3$ достигается при $z \geqslant \beta$, а при $n = 4$ требуется как минимум две итерации.

Из тех же соображений, что и в доказательстве теоремы 1, можно видеть, что максимально возможное значение для $D(k+1)$ достигается, когда после первой итерации получили множество J мощности $D(k)$, и при этом $D(k+1) = 2 \cdot D(k)$ (что выполняется при $z < \beta$). Значит, действует рекуррентное соотношение

$$D(k+1) = 2 \cdot D(k), \quad k \geqslant 1.$$

В замкнутой форме это соотношение примет вид

$$D(k) = 3 \cdot 2^{k-1}, \quad k \geqslant 1.$$

Теорема 2 доказана.

3.2. Наибольшее и наименьшее число итераций для данного размера множества. Выразим количество итераций через функцию от n и установим взаимосвязь между введёнными величинами.

Определение 2. Положим

$$k_{\min}(n) := \underset{k}{\operatorname{argmin}}\{D(k) \mid D(k) \geqslant n\},$$
$$k_{\max}(n) := \underset{k}{\operatorname{argmax}}\{d(k) \mid d(k) \leqslant n\}.$$

Теорема 3. *Справедливы неравенства*

$$k_{\min}(n) \leqslant \widehat{k} \leqslant k_{\max}(n). \tag{11}$$

Доказательство. 1) Если предположить, что $\widehat{k} < k_{\min}(n)$, то по определению $k_{\min}$ и с учётом монотонного возрастания функции D имеем $D(\widehat{k}) < n$, что противоречит (7). Следовательно, $k_{\min}(n) \leqslant \widehat{k}$.

2) Аналогично показывается, что $\widehat{k} \leqslant k_{\max}(n)$. Теорема 3 доказана.

Из (11) следует, что число возможных значений для количества итераций $\widehat{k}$ равно $k_{\max}(n) - k_{\min}(n) + 1$.

Кроме того, из определения $k_{\min}$ и $k_{\max}$ следует, что при условии $k_{\min}(n) \geqslant 2$ выполняются неравенства

$$D(k_{\min}(n) - 1) < n \leqslant D(k_{\min}(n)), \tag{12}$$

$$d(k_{\max}(n)) \leqslant n < d(k_{\max}(n) + 1), \tag{13}$$

стало быть,

$$d(k_{\max}(n)) \leqslant n \leqslant D(k_{\min}(n)). \tag{14}$$

Табл. 1 содержит несколько первых значений d и D, которые можно посчитать по формулам (8) и (10).

Таблица 1

k	1	2	3	4	5	6	7	8	9	10
$d(k)$	1	3	4	6	10	18	34	66	130	258
$D(k)$	3	6	12	24	48	96	192	384	768	1536

Для примера рассмотрим $n = 100$. Минимальное значение функции $D(k)$, которое не меньше 100, это $D(7) = 192$. Следовательно, $k_{\min}(n) = 7$. В то же время, максимальное значение для $d(k)$, которое не больше 100, это $d(8) = 66$. Значит, $k_{\max}(n) = 8$. Таким образом, любая задача при $n = 100$ потребует 7 или 8 итераций алгоритма Maculan.

3.3. Нахождение $k_{\min}(n)$ и $k_{\max}(n)$. Ниже нам потребуется гарантировать, что $k_{\max}(n) \geqslant 2$ и $k_{\min}(n) \geqslant 2$. Из табл. 1 и определения $k_{\min}$ видно, что оба условия выполняются при $n \geqslant 4$.

Вернёмся к формулам (12) и (13). Из них и полученных для функций $D(k)$ и $d(k)$ представлений (8) и (10) следует, что при условии $n \geqslant 4$ выполняются неравенства

$$3 \cdot 2^{k_{\min}(n)-2} < n \leqslant 3 \cdot 2^{k_{\min}(n)-1}, \tag{15}$$

$$2^{k_{\max}(n)-2} + 2 \leqslant n < 2^{k_{\max}(n)-1} + 2. \tag{16}$$

Преобразовав эти неравенства, можно получить оценки для значений $k_{\min}(n)$ и $k_{\max}(n)$ через n в явном виде. Для начала найдём выражение для $k_{\min}(n)$.

Теорема 4. *Пусть $n \geqslant 4$. Тогда*

$$\log_2(n) - \log_2 3 + 1 \leqslant k_{\min}(n) < \log_2(n) - \log_2 3 + 2. \tag{17}$$

Доказательство. Прологарифмируем неравенства (15) по основанию 2:

$$\log_2 3 + k_{\min}(n) - 2 < \log_2(n) \leqslant \log_2 3 + k_{\min}(n) - 1.$$

Вычтем из всех частей этой формулы значение $k_{\min}(n)$ и домножим всё на -1. Получим

$$1 - \log_2 3 \leqslant k_{\min}(n) - \log_2(n) < 2 - \log_2 3.$$

Наконец, добавив ко всем частям $\log_2(n)$, придём к соотношениям (17). Теорема 4 доказана.

Следствие 1. *Учитывая, что* $\log_2 3 \approx 1.585$, *оценку* (17) *можно переписать в следующем виде:*

$$\log_2(n) - 0.585 < k_{\min}(n) < \log_2(n) + 0.415. \tag{18}$$

Следствие 2. *Левая и правая части* (17) *отличаются ровно на единицу, и одно из неравенств строгое. С учётом этого и поскольку* $k_{\min}(n)$ *— целое число, можно сделать вывод, что*

$$k_{\min}(n) = \lceil \log_2(n) - \log_2 3 + 1 \rceil, \tag{19}$$

где $\lceil x \rceil = \min\{n \mid n \in \mathbb{Z},\, n \geqslant x\}$ *— наименьшее большее целое для* x.

Найдём оценки для $k_{\max}(n)$.

Теорема 5. *Пусть* $n \geqslant 4$. *Справедливы неравенства*

$$\log_2(n) < k_{\max}(n) < \log_2(n) + 2. \tag{20}$$

Доказательство. Прологарифмируем неравенства (16) по основанию 2:

$$\begin{aligned} k_{\max}(n) - 2 + \log_2(1 + 2^{3-k_{\max}(n)}) &\leqslant \log_2(n) \\ &< k_{\max}(n) - 1 + \log_2(1 + 2^{2-k_{\max}(n)}). \end{aligned}$$

Вычтем из всех частей неравенства значение $k_{\max}(n)$ и домножим всё на -1. Получим

$$\begin{aligned} 1 - \log_2(1 + 2^{2-k_{\max}(n)}) &< k_{\max}(n) - \log_2(n) \\ &\leqslant 2 - \log_2(1 + 2^{3-k_{\max}(n)}), \end{aligned} \tag{21}$$

откуда следует, что

$$k_{\max}(n) = \log_2(n) + \Delta,$$

где

$$\Delta \in (1 - \log_2(1 + 2^{2-k_{\max}(n)}), 2 - \log_2(1 + 2^{3-k_{\max}(n)})].$$

При сохранении условия $k_{\max}(n) \geqslant 2$ левая граница этого интервала изменяется в пределах от 0 до $1-\varepsilon$, а правая — в пределах от примерно 0.415 до $2-\varepsilon$, где $\varepsilon > 0$ — сколь угодно малая величина. Таким образом, упрощённая форма условий (21) будет выглядеть так:

$$\log_2(n) < k_{\max}(n) < \log_2(n) + 2.$$

Теорема 5 доказана.

Теперь объединим оценки (18) и (20).

Следствие 3. *Из неравенств* (11) *вытекает, что для* $n \geqslant 4$ *количество итераций алгоритма Maculan лежит в пределах*

$$\log_2(n) - 0.585 < \widehat{k} < \log_2(n) + 2. \tag{22}$$

Сформулируем основной результат разд. 3.

Теорема 6. *Пусть* $n \geqslant 1$. *Для любого множества* $\{c_1, \ldots, c_n\}$ *при условии, что* $c_i \neq c_j$ *для всех* $i \neq j$, *число итераций алгоритма Maculan лежит в пределах трёх возможных значений:* k, $k+1$ *и* $k+2$, *где*

$$k = \lceil \log_2(n) - \log_2 3 + 1 \rceil.$$

Доказательство. Прибавим неравенство $\log_2(n) - 0.585 < k_{\min}(n)$ из (18) к неравенству $k_{\max}(n) < \log_2(n) + 2$ из (20). Получим

$$k_{\max}(n) - k_{\min}(n) < \log_2(n) + 2 - \log_2(n) + 0.585 = 2.585.$$

Поскольку разность целых чисел есть целое число, то

$$k_{\max}(n) - k_{\min}(n) \leqslant 2. \tag{23}$$

Таким образом, $k_{\max}$ не превосходит $k_{\min} + 2$. Остаётся лишь добавить к этому выражение для $k_{\min}$ из формулы (19). Теорема 6 доказана.

Результат теоремы 6 позволяет заключить, что в случае неравенства компонент все входные данные с одинаковым n практически эквивалентны по числу итераций алгоритма. Таким образом, можно считать случай неравных компонент худшим случаем для алгоритма Maculan.

4. Численные эксперименты

Выше отмечалось, что задача поиска ортогональной проекции точки $c = (c_1, \ldots, c_n)$ на стандартный симплекс сводится к решению уравнения (2) при β, равном 1.

Были проведены массовые эксперименты проецирования точки на стандартный симплекс с использованием реализации алгоритма Maculan на языке Java. Брали 10000 точек в n-мерном евклидовом пространстве при n, равном 10, 10^2, 10^3, 10^4, 10^5 и 10^6.

Координаты точек $c = (c_1, \ldots, c_n)$ формировались тремя способами:

(A) генерировались по непрерывному равномерному распределению на отрезке $[-10000, 10000]$;

(B) $c_j \neq c_\ell$ для всех $1 \leqslant j \neq \ell \leqslant n$, т. е. все компоненты точки c различны;

(C) $c_j = \xi$ для всех $1 \leqslant j \leqslant n$, где ξ — произвольное фиксированное число из отрезка $[-10000, 10000]$.

В целом, численные результаты подтверждают теоретические выводы. В приведённой ниже табл. 2 указано время в миллисекундах. В случаях (A) и (B) алгоритм Maculan показывает практически одинаковые результаты. Это объясняется тем, что при равномерном распределении компоненты вектора c генерируются случайным образом и их совпадение крайне маловероятно. В случае (C) (равенства всех компонент вектора c) Maculan работает в разы быстрее, чем в других случаях.

Таблица 2

случай \ n	10	10^2	10^3	10^4	10^5	10^6
(A)	17	151	1302	13192	140142	1825864
(B)	19	145	1311	13380	140653	1828122
(C)	6	43	374	3875	43410	640624

5. Заключение

В работе проведён подробный разбор алгоритма из [30] по поиску нуля выпуклой кусочно-линейной функции. Время работы алгоритма оценено как с теоретической точки зрения, так и с помощью численных экспериментов. Теоретически показано, что хотя общая оценка сложности алгоритма равна $O(n)$, можно выделить два крайних типа входных данных. В случае равенства всех компонент c_j входного множества S алгоритм всегда завершает работу после одной итерации. С другой стороны, в случае, когда все значения c_j различны, алгоритм совершает практически максимальное число итераций для данной мощности n множества S и это значение известно с точностью до трёх итераций. В промежуточных случаях (когда совпадает часть элементов c_j) число итераций алгоритма может варьироваться в зависимости от количества и расположения повторяющихся элементов. Практические испытания подтвердили теоретические заключения. Из табл. 2 видно, что время работы на множествах из равных элементов в разы меньше, чем время работы на случайных данных.

ЛИТЕРАТУРА

1. **Ахо А., Хопкрофт Дж. Э., Ульман Дж. Д.** Структуры данных и алгоритмы. М.: Издат. дом Вильямс, 2003. 384 с.
2. **Величко А. С.** О выборе шага в проективных алгоритмах для задач линейного программирования большой размерности // Дальневост. мат. журн. 2012. № 2. С. 160–170.
3. **Демьянов В. Ф., Тамасян Г. Ш.** О прямых методах решения вариационных задач // Тр. Ин-та математики и механики УрО РАН. 2010. Т. 16, № 5. С. 36–47.
4. **Долгий Д. В., Нурминский Е. А.** Ускоренный параллельный метод проекций для решения задачи о наименьшем расстоянии // Вычисл. методы и программирование. 2006. Т. 7, вып. 3. С. 273–277.
5. **Долгополик М. В., Тамасян Г. Ш.** Об эквивалентности методов наискорейшего и гиподифференциального спусков в некоторых задачах условной оптимизации // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2014. Т. 14, вып. 4, ч. 2. С. 532–542.
6. **Зоркальцев В. И.** Октаэдрические и евклидовы проекции точки на линейное многообразие // Тр. Ин-та математики и механики УрО РАН. 2012. Т. 18, № 3. С. 106–118.
7. **Зоркальцев В. И.** Проекции точки на полиэдр // Журн. вычисл. математики и мат. физики. 2013. Т. 53, № 1. С. 4–19.
8. **Малоземов В. Н.** МДМ-методу — 40 лет // Вестн. Сыктывкар. ун-та. Сер. 1. 2012. Вып. 15. С. 51–62.
9. **Малоземов В. Н., Певный А. Б.** Быстрый алгоритм проектирования точки на симплекс // Вестн. СПбГУ. Сер. 1. 1992. № 1. С. 112–113.
10. **Малоземов В. Н., Тамасян Г. Ш.** Два быстрых алгоритма проектирования точки на стандартный симплекс // Журн. вычисл. математики и мат. физики. 2016. Т. 56, № 5. С. 742—755.
11. **Малоземов В. Н., Тамасян Г. Ш.** Лемма Гиббса и её приложения // Семинар «Constructive Nonsmooth Analysis and Nondifferentiable Optimization»: Избр. докл. http://www.apmath.spbu.ru/cnsa/pdf/2017/LemmaGibbsa.pdf
12. **Митчелл Б. Ф., Демьянов В. Ф., Малоземов В. Н.** Нахождение ближайшей к началу координат точки многогранника // Вестн. ЛГУ. 1971. № 19. С. 38–45.
13. **Нурминский Е. А.** Параллельный метод проекции на выпуклую оболочку семейства множеств // Изв. вузов. Математика. 2003. № 12. С. 78–82.

14. **Нурминский Е. А.** Проекция на внешне заданные полиэдры // Журн. вычисл. математики и мат. физики. 2008. Т. 48, № 3. С. 387–396.
15. **Тамасян Г. Ш.** О методах наискорейшего и гиподифференциального спуска в одной задаче вариационного исчисления // Вычисл. методы и программирование. 2012. Т. 13, № 1. С. 197–217.
16. **Тамасян Г. Ш.** Численные методы в задачах вариационного исчисления для функционалов, зависящих от производных высшего порядка // Пробл. мат. анализа. Вып. 67. Новосибирск: Изд-во Тамара Рожковская, 2012. С. 113–132.
17. **Тамасян Г. Ш., Просолупов Е. В., Ангелов Т. А.** Сравнительный анализ двух быстрых алгоритмов проецирования точки на стандартный симплекс // Дискрет. анализ и исслед. операций. 2016. Т. 23, № 2. С. 100–123.
18. **Тамасян Г. Ш., Чумаков А. А.** Нахождение расстояния между эллипсоидами // Дискрет. анализ и исслед. операций. 2014. Т. 21, № 3. С. 87–102.
19. **Утешев А. Ю., Яшина М. В.** Нахождение расстояния от эллипсоида до плоскости и квадрики в $\mathbb{R}^n$ // Докл. АН. 2008. Т. 419, № 4. С. 471–474.
20. **Brucker P.** An $O(n)$ algorithm for quadratic knapsack problems // Oper. Res. Lett. 1984. Vol. 3, No. 3. P. 163–166.
21. **Causa A., Raciti F.** A purely geometric approach to the problem of computing the projection of a point on a simplex // J. Optim. Theory Appl. 2013. Vol. 156, No. 2. P. 524–528.
22. **Demyanov V. F.** Algorithms for some minimax problems // J. Comput. Syst. Sci. 1968. Vol. 2, No. 4. P. 342–380.
23. **Demyanov V. F., Giannessi F., Tamasyan G. Sh.** Variational control problems with constraints via exact penalization // Variational Analysis and Applications. New York: Springer-Verl., 2005. P. 301–342.
24. **Demyanov V. F., Tamasyan G. Sh.** Exact penalty functions in isoperimetric problems // Optimization. 2011. Vol. 60, No. 1–2. P. 153–177.
25. **Demyanov V. F., Tamasyan G. Sh.** Direct methods in the parametric moving boundary variational problem // Numer. Funct. Anal. Optimization. 2014. Vol. 35, No. 7–9. P. 932–961.
26. **Deutsch F.** The method of alternating orthogonal projections // Approximation Theory, Spline Functions and Applications. Dordrecht: Kluwer Acad. Publ., 1992. P. 105–121.
27. **Dolgopolik M. V., Tamasyan G. Sh.** Method of steepest descent for two-dimensional problems of calculus of variations // Constructive Nonsmooth Analysis and Related Topics. New York: Springer-Verl., 2014. P. 101–113. (Optim. Its Appl.; Vol. 87).
28. **Held M., Wolfe P., Crowder H. P.** Validation of the subgradient optimization // Math. Program. 1974. Vol. 6, No. 1. P. 62–88.

29. **Helgason R. V., Kennington J. L., Lall H.** A polynomially bounded algorithm for a singly constrained quadratic program // Math. Program. 1980. Vol. 18, No. 1. P. 338–343.
30. **Maculan N., Galdino de Paula G., Jr.** A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$ // Oper. Res. Lett. 1989. Vol. 8, No. 4. P. 219–222.
31. **Michelot C.** A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$ // J. Optim. Theory Appl. 1986. Vol. 50, No. 1. P. 195–200.
32. **Patriksson M.** A survey on the continuous nonlinear resource allocation problem // Eur. J. Oper. Res. 2008. Vol. 185, No. 1. P. 1–46.
33. **Tamasyan G., Chumakov A.** Finding the distance between the ellipsoid and the intersection of a linear manifold and ellipsoid // Proc. 2015 Int. Conf. Stability and Control Processes in Memory of V. I. Zubov joined with 21st Int. Workshop on Beam Dynamics and Optimization (St. Petersburg, Russia, Oct. 5–9, 2015). Piscataway: IEEE, 2015. P. 357–360.
34. **Tamasyan G., Prosolupov E.** Orthogonal projection of a point onto the standard simplex algorithms analysis // Proc. 2015 Int. Conf. Stability and Control Processes in Memory of V. I. Zubov joined with 21st Int. Workshop on Beam Dynamics and Optimization (St. Petersburg, Russia, Oct. 5–9, 2015). Piscataway: IEEE, 2015. P. 353–356.

Просолупов Евгений Викторович,
Тамасян Григорий Шаликович

Статья поступила
10 марта 2017 г.

Исправленный вариант —
26 декабря 2017 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
April–June 2018. Volume 25, No. 2. P. 82–100

UDC 519.8 DOI: 10.17377/daio.2018.25.571

COMPLEXITY ESTIMATION FOR AN ALGORITHM OF SEARCHING FOR ZERO OF A PIECEWISE LINEAR CONVEX FUNCTION

E. V. Prosolupov[a] and *G. Sh. Tamasyan*[b]
St. Petersburg State University,
Universitetskii Ave., 35, 198504 St. Petersburg, Russia
E-mail: [a]e.prosolupov@spbu.ru, [b]g.tamasyan@spbu.ru

Abstract. It is known that the problem of the orthogonal projection of a point to the standard simplex can be reduced to solution of a scalar equation. In this article, the complexity is analyzed of an algorithm of searching for zero of a piecewise linear convex function which is proposed by N. Maculan and G. Galdino de Paula, Jr. (Oper. Res. Lett. **8** (4), 219–222 (1989)). The analysis is carried out of the best and worst cases of the input data for the algorithm. To this end, the largest and smallest numbers of iterations of the algorithm are studied as functions of the size of the input data. It is shown that, in the case of equality of elements of the input set, the algorithm performs the smallest number of iterations. In the case of different elements of the input set, the number of iterations is maximal and depends rather weakly on the particular values of the elements of the set. The results of numerical experiments with random input data of large dimension are presented. Tab. 2, illustr. 2, bibliogr. 34.

Keywords: standard simplex, orthogonal projection of a point, zeros of function.

REFERENCES

1. **A. V. Aho, J. E. Hopcroft,** and **J. D. Ullman,** *Data Structures and Algorithms,* Addison-Wesley, Reading, MA, 1983. Translated under the title *Struktury dannykh i algoritmy,* Izd. dom Williams, Moscow, 2003 [Russian].
2. **A. S. Velichko,** On the step choice in projection algorithms for large-scale linear programming problems, *Dal'nevost. Mat. Zh.,* **12**, No. 2, 160–170, 2012 [Russian].
3. **V. F. Demyanov** and **G. Sh. Tamasyan,** On direct methods for solving variational problems, *Tr. Inst. Mat. Mekh.,* **16**, No. 5, 36–47, 2010 [Russian].
4. **D. V. Dolgii, E. A. Nurminskii,** An accelerated parallel projection method for solving the minimum length problem, *Vychisl. Metody Program.,* **7**, No. 3, 273–277, 2006 [Russian].

5. **M. V. Dolgopolik** and **G. Sh. Tamasyan,** On equivalence of the method of steepest descent and the method of hypodifferential descent in some constrained optimization problems, *Izv. Sarat. Univ., Ser. Mat. Mekh. Inform.,* **14**, No. 4-2, 532–542, 2014 [Russian].
6. **V. I. Zorkal'tsev,** Octahedral and Euclidean projections of a point to a linear manifold, *Tr. Inst. Mat. Mekh.,* **18**, No. 3, 106–118, 2012 [Russian]. Translated in *Proc. Steklov Inst. Math.*, **284**, Suppl. 1, 185–197, 2014.
7. **V. I. Zorkal'tsev,** Projecting a point on a polyhedron, *Zh. Vychisl. Mat. Mat. Fiz.,* **53**, No. 1, 4–19, 2013 [Russian].
8. **V. N. Malozemov,** MDM method — 40 years, *Vestn. Syktyvkar. Univ., Ser. 1,* No. 15, 51–62, 2012 [Russian].
9. **V. N. Malozemov** and **A. B. Pevnyi,** Fast algorithm for projecting a point on a simplex, *Vestn. St-Peterbg. Univ., Ser. 1,* No. 1, 112–113, 1992 [Russian]. Translated in *Vestn. St. Petersbg. Univ., Math.,* **25**, No. 1, 62–63, 1992.
10. **V. N. Malozemov** and **G. Sh. Tamasyan,** Two fast algorithms for finding the projection of a point onto the standard simplex, *Zh. Vychisl. Mat. Mat. Fiz.,* **56**, No. 5, 742–755, 2016 [Russian]. Translated in *Comput. Math. Math. Phys.,* **56**, No. 5, 730–743, 2016.
11. **V. N. Malozemov** and **G. Sh. Tamasyan,** Gibb's lemma and its applications, *Seminar on Constructive Nonsmooth Analysis and Nondifferentiable Optimization, Selected Talks,* St-Peterbg. Gos. Univ., St. Petersburg, 2017. Available at http://www.apmath.spbu.ru/cnsa/pdf/2017/LemmaGibbsa.pdf (accessed Jan. 3, 2018) [Russian].
12. **B. F. Mitchell, V. F. Demyanov,** and **V. N. Malozemov,** Finding point of polyhedron nearest to the origin, *Vestn. Leningr. Univ.,* No. 19, 38–45, 1971 [Russian].
13. **E. A. Nurminski,** A parallel method of projection onto the convex hull of a family of sets, *Izv. Vyssh. Uchebn. Zaved., Mat.,* No. 12, 78–82, 2003 [Russian]. Translated in *Russ. Math.,* **47**, No. 12, 74–78, 2003.
14. **E. A. Nurminski,** Projection onto polyhedra in outer representation, *Zh. Vychisl. Mat. Mat. Fiz.,* **48**, No. 3, 387–396, 2008 [Russian]. Translated in *Comput. Math. Math. Phys.,* **48**, No. 3, 367–375, 2008.
15. **G. Sh. Tamasyan,** Methods of steepest and hypodifferential descent in one problem of calculus of variations, *Vychisl. Metody Program.,* **13**, No. 1, 197–217, 2012 [Russian].
16. **G. Sh. Tamasyan,** Numerical methods in problems of calculus of variations for functionals depending on higher order derivatives, in *Problems of Mathematical Analysis,* Vol. 67, pp. 113–132, Tamara Rozhkovskaya, Novosibirsk, 2012 [Russian]. Translated in *J. Math. Sci.,* **188**, No. 3, 299–321, 2013.
17. **G. Sh. Tamasyan, E. V. Prosolupov,** and **T. A. Angelov,** Comparative study of two fast algorithms for projecting a point to the standard simplex, *Diskretn. Anal. Issled. Oper.,* **23**, No. 2, 100–123, 2016 [Russian]. Translated in *J. Appl. Ind. Math.,* **10**, No. 2, 288–301, 2016.

18. **G. Sh. Tamasyan** and **A. A. Chumakov,** Finding the distance between ellipsoids, *Diskretn. Anal. Issled. Oper.,* **21**, No. 3, 87–102, 2014 [Russian]. Translated in *J. Appl. Ind. Math.,* **8**, No. 3, 400–410, 2014.
19. **A. Yu. Uteshev** and **M. V. Yashina,** Computation of the distance from an ellipsoid to a linear surface and a quadric in $\mathbb{R}^n$, *Dokl. Akad. Nauk,* **419**, No. 4, 471–474, 2008 [Russian]. Translated in *Dokl. Math.,* **77**, No. 2, 269–272, 2008.
20. **P. Brucker,** An $O(n)$ algorithm for quadratic knapsack problems, *Oper. Res. Lett.,* **3**, No. 3, 163–166, 1984.
21. **A. Causa** and **F. Raciti,** A purely geometric approach to the problem of computing the projection of a point on a simplex, *J. Optim. Theory Appl.,* **156**, No. 2, 524–528, 2013.
22. **V. F. Demyanov,** Algorithms for some minimax problems, *J. Comput. Syst. Sci.,* **2**, No. 4, 342–380, 1968.
23. **V. F. Demyanov, F. Giannessi,** and **G. Sh. Tamasyan,** Variational control problems with constraints via exact penalization, in *Variational Analysis and Applications,* pp. 301–342, Springer, New York, 2005 (Nonconvex Optim. Its Appl., Vol. 79).
24. **V. F. Demyanov** and **G. Sh. Tamasyan,** Exact penalty functions in isoperimetric problems, *Optim.,* **60**, No. 1–2, 153–177, 2011.
25. **V. F. Demyanov** and **G. Sh. Tamasyan,** Direct methods in the parametric moving boundary variational problem, *Numer. Funct. Anal. Optim.,* **35**, No. 7–9, 932–961, 2014.
26. **F. Deutsch,** The method of alternating orthogonal projections, in *Approximation Theory, Spline Functions and Applications,* pp. 105–121, Kluwer Acad. Publ., Dordrecht, 1992.
27. **M. V. Dolgopolik** and **G. Sh. Tamasyan,** Method of steepest descent for two-dimensional problems of calculus of variations, in *Constructive Nonsmooth Analysis and Related Topics,* pp. 101–113, Springer, New York, 2014 (Springer Optim. Its Appl., Vol. 87).
28. **M. Held, P. Wolfe,** and **H. P. Crowder,** Validation of subgradient optimization, *Math. Program.,* **6**, No. 1, 62–88, 1974.
29. **R. V. Helgason, J. L. Kennington,** and **H. Lall,** A polynomially bounded algorithm for a singly constrained quadratic program, *Math. Program.,* **18**, No. 1, 338–343, 1980.
30. **N. Maculan** and **G. Galdino de Paula, Jr.,** A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$, *Oper. Res. Lett.,* **8**, No. 4, 219–222, 1989.
31. **C. Michelot,** A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$, *J. Optim. Theory Appl.,* **50**, No. 1, 195–200, 1986.
32. **M. Patriksson,** A survey on the continuous nonlinear resource allocation problem, *Eur. J. Oper. Res.,* **185**, No. 1, 1–46, 2008.

33. **G. Sh. Tamasyan** and **A. A. Chumakov,** Finding the distance between an ellipsoid and the intersection of a linear manifold and ellipsoid, *Proc. 2015 Int. Conf. "Stability and Control Processes" in Memory of V. I. Zubov, St. Petersburg, Russia, Oct. 5–9, 2015,* pp. 357–360, IEEE, Piscataway, 2015.
34. **G. Sh. Tamasyan** and **E. V. Prosolupov,** Orthogonal projection of a point onto the standard simplex algorithms analysis, *Proc. 2015 Int. Conf. "Stability and Control Processes" in Memory of V. I. Zubov, St. Petersburg, Russia, Oct. 5–9, 2015,* pp. 353–356, IEEE, Piscataway, 2015.

Evgeny V. Prosolupov,
Grigory Sh. Tamasyan

Received
10 March 2017

Revised
26 December 2017