КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ И МОДЕЛИРОВАНИЕ 2022 Т. 14 № 2 С. 213–223
DOI: 10.20537/2076-7633-2022-14-2-213-223

Ки&М

МАТЕМАТИЧЕСКИЕ ОСНОВЫ И ЧИСЛЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ

УДК: 519.85

Нижние оценки для методов типа условного градиента для задач минимизации гладких сильно выпуклых функций

А. Д. Агафонов

Национальный исследовательский университет «Московский физико-технический институт»,
Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., 9

E-mail: agafonov.ad@phystech.edu

Получено 10.03.2020, после доработки — 12.03.2022.
Принято к публикации 14.03.2022.

В данной работе рассматриваются методы условного градиента для оптимизации сильно выпуклых функций. Это методы, использующие линейный минимизационный оракул, то есть умеющие вычислять решение задачи

$$\operatorname*{Argmin}_{x\in X}\langle p,\ x\rangle$$

для заданного вектора $p \in \mathbb{R}^n$. Существует целый ряд методов условного градиента, имеющих линейную скорость сходимости в сильно выпуклом случае. Однако во всех этих методах в оценку скорости сходимости входит размерность задачи, которая в современных приложениях может быть очень большой. В данной работе доказывается, что в сильно выпуклом случае скорость сходимости методов условного градиента в лучшем случае зависит от размерности задачи n как $\widetilde{\Omega}\left(\sqrt{n}\right)$. Таким образом, методы условного градиента могут оказаться неэффективными для решения сильно выпуклых оптимизационных задач больших размерностей.

Отдельно рассматривается приложение методов условного градиента к задачам минимизации квадратичной формы. Уже была доказана эффективность метода Франк – Вульфа для решения задачи квадратичной оптимизации в выпуклом случае на симплексе (PageRank). Данная работа показывает, что использование методов условного градиента для минимизации квадратичной формы в сильно выпуклом случае малоэффективно из-за наличия размерности в оценке скорости сходимости этих методов. Поэтому рассматривается метод рестартов условного градиента (Shrinking Conditional Gradient). Его отличие от методов условного градиента заключается в том, что в нем используется модифицированный линейный минимизационный оракул, который для заданного вектора $p \in \mathbb{R}^n$ вычисляет решение задачи

$$\operatorname{Argmin}\{\langle p,\ x\rangle\colon\ x \in X,\ \|x - x_0\| \leqslant R\}.$$

В оценку скорости сходимости такого алгоритма размерность уже не входит. С помощью рестартов метода условного градиента получена сложность (число арифметических операций) минимизации квадратичной формы на ∞-шаре. Полученная оценка работы метода сравнима со сложностью градиентного метода.

Ключевые слова: метод Франк – Вульфа, рестарты

Исследование выполнено за счет гранта Российского научного фонда (проект № 21-71-30005).

COMPUTER RESEARCH AND MODELING
2022 VOL. 14 NO. 2 P. 213–223
DOI: 10.20537/2076-7633-2022-14-2-213-223

Ки&М

MATHEMATICAL MODELING AND NUMERICAL SIMULATION

UDC: 519.85

Lower bounds for conditional gradient type methods for minimizing smooth strongly convex functions

A. D. Agafonov

National Research University Moscow Institute of Physics and Technology,
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia

E-mail: agafonov.ad@phystech.edu

Received 10.03.2020, after completion — 12.03.2022.
Accepted for publication 14.03.2022.

In this paper, we consider conditional gradient methods for optimizing strongly convex functions. These are methods that use a linear minimization oracle, which, for a given vector $p \in \mathbb{R}^n$, computes the solution of the subproblem

$$\operatorname*{Argmin}_{x \in X} \langle p, x \rangle.$$

There are a variety of conditional gradient methods that have a linear convergence rate in a strongly convex case. However, in all these methods, the dimension of the problem is included in the rate of convergence, which in modern applications can be very large. In this paper, we prove that in the strongly convex case, the convergence rate of the conditional gradient methods in the best case depends on the dimension of the problem n as $\widetilde{\Omega}\left(\sqrt{n}\right)$. Thus, the conditional gradient methods may turn out to be ineffective for solving strongly convex optimization problems of large dimensions.

Also, the application of conditional gradient methods to minimization problems of a quadratic form is considered. The effectiveness of the Frank – Wolfe method for solving the quadratic optimization problem in the convex case on a simplex (PageRank) has already been proved. This work shows that the use of conditional gradient methods to solve the minimization problem of a quadratic form in a strongly convex case is ineffective due to the presence of dimension in the convergence rate of these methods. Therefore, the Shrinking Conditional Gradient method is considered. Its difference from the conditional gradient methods is that it uses a modified linear minimization oracle. It's an oracle, which, for a given vector $p \in \mathbb{R}^n$, computes the solution of the subproblem

$$\operatorname{Argmin}\{\langle p, x \rangle \colon x \in X, \ \|x - x_0\| \leqslant R\}.$$

The convergence rate of such an algorithm does not depend on dimension. Using the Shrinking Conditional Gradient method the complexity (the total number of arithmetic operations) of solving the minimization problem of quadratic form on a ∞-ball is obtained. The resulting evaluation of the method is comparable to the complexity of the gradient method.

Keywords: Frank – Wolfe method, Shrinking Conditional Gradient

Citation: *Computer Research and Modeling*, 2022, vol. 14, no. 2, pp. 213–223 (Russian).
This research was funded by the Russian Science Foundation (project 21-71-30005).

Введение

В данной работе рассматриваются методы оптимизации, использующие линейный минимизационный оракул. Это такой оракул, который для данного вектора $p \in \mathbb{R}^n$ вычисляет решение задачи

$$\operatorname*{Argmin}_{x \in X} \langle p,\ x \rangle.$$

Такими методами являются классический метод условного градиента (Франк – Вульфа) [Frank, Wolfe, 1956] и различные его варианты [Jaggi, 2013]. Эти методы — одни из самых популярных методов условной оптимизации вследствие их простоты и отсутствия проектирования на шаге алгоритма [Dvurechensky et al., 2020]. В основе метода условного градиента лежат идея линеаризации целевой функции в точке и минимизация линейной аппроксимации на бюджетном множестве. Однако этот метод сходится достаточно медленно, сублинейно ($O(1/\varepsilon)$) [Левитин, Поляк, 1966].

Ряд недавних работ показал, что в случае сильно выпуклых функций оценку можно улучшить до линейной ($O(\log 1/\varepsilon)$) [Garber, Hazan, 2013; Kerdreux, d'Aspremont, Pokutta, 2019; Lacoste-Juien, Jaggi, 2015]. Однако во всех алгоритмах, представленных в работах выше, в оценку скорости сходимости входит размерность задачи (см. дискуссию в [Lacoste-Juien, Jaggi, 2015]). При ряде дополнительных предположений наличия размерности в оценке можно избежать [Carderera et al., 2021]. Стоит отметить, что такая проблема возникает только в сильно выпуклом случае. Например, в оценку скорости сходимости классического метода Франк – Вульфа, который работает для выпуклых задач, размерность не входит.

В данной работе мы показываем, что для задачи сильно выпуклой оптимизации в оценку скорости сходимости методов, использующих линейный минимизационный оракул, обязательно входит размерность задачи как $\widetilde{\Omega}\left(\sqrt{n}\right)$. В современных приложениях размерность может быть очень большой, что негативно сказывается на времени работы метода. Таким образом, методы условного градиента могут оказаться неэффективными для решения задач больших размерностей в сильно выпуклом случае.

Также рассматривается приложение методов условного градиента к задаче квадратичной оптимизации на единичном шаре в ∞-норме. Такая постановка задачи на ∞-шаре или, что эквивалентно, параллелепипеде встречается достаточно часто [Горнов, 2009]. Например, она появляется в задачах, где известно, что каждая компонента ограниченна, то есть принадлежит заданному отрезку. В данной работе рассматривается случай, когда матрица A дважды разрежена (одновременно разрежена по столбцам и строкам) [Аникин и др., 2015]. Если считать, что матрица A имеет размер $n \times n$, а число элементов в каждой строке и столбце не больше чем $s \ll n$, то число ненулевых элементов в матрице может быть sn.

Было показано [Аникин и др., 2015; Гасников, 2016], что классический метод Франк – Вульфа эффективно решает задачу минимизации квадратичной формы на симплексе (PageRank [Brin, Page, 1998]). В данной работе рассматривается возможность решения задачи квадратичной оптимизации методами типа условного градиента в сильно выпуклом случае. Алгоритмы, использующие линейный минимизационный оракул, не могут решать эту задачу эффективно. В данной работе доказывается, что оценки скорости сходимости таких методов содержат размерность. Для решения задачи минимизации квадратичной формы авторами используется метод рестартов условного градиента (Shrinking Conditional Gradient) [Lan, 2013]. Вместо линейного минимизационного оракула используется оракул, вычисляющий решение задачи

$$\operatorname{Argmin}\{\langle p,\ x \rangle \colon\ x \in X,\ \|x - x_0\| \leqslant R\}.$$

В сложность этого метода размерность уже не входит. Анализ работы этого метода показывает, что сложность метода (число арифметических операций) равна

$$\frac{8L}{\mu}O(n) + \frac{8L}{\mu}\left(sn\log\frac{\mu R_0}{\varepsilon}\right),$$

где первый член — затраты на препроцессинг. Эта оценка сравнима со сложностью градиентного метода [Allen-Zhu, Orecchia, 2014].

Статья состоит из введения и трех основных разделов. В первом разделе рассматривается классический метод условного градиента (Франк – Вульфа) и вводятся необходимые вспомогательные понятия. Второй раздел посвящен вопросам размерности в оценке скорости сходимости методов условного градиента. В третьем разделе предлагается подход к решению задачи минимизации квадратичной формы на ∞-шаре с помощью рестартов метода условного градиента.

Классический метод условного градиента

Рассмотрим следующую задачу оптимизации:

$$f(x) \to \min_{x\in X}, \tag{1}$$

где

- $X \in \mathbb{R}^n$ — выпуклое и замкнутое множество,
- $f(x)$ — выпуклая функция,
- $f(x)$ ограничена снизу на X и достигает своего минимума в x^*.

Рассмотрим классический метод условного градиента [Frank, Wolfe, 1956]. В его основе лежит идея линеаризации функции. На k-м шаге метода в точке x_k линеаризуем функцию $f(x)$, минимизируем эту линейную аппроксимацию на X и найденную точку используем как направление движения.

Алгоритм 1. Класический метод условного градиента (Франк – Вульфа)

1: **Дано:** $x_0 \in X$ — начальная точка. Положим $y_0 = x_0$.
2: **for** $k = 1, 2, \ldots$ **do**
3: Вычисляем $x_k = \underset{x\in X}{\operatorname{argmin}}\,\langle f'(y_{k-1}),\ x\rangle$.
4: Положим $y_k = y_{k-1} + (1-\gamma_k)x_k$, где $\gamma_k \in [0,\ 1]$.
5: **end for**

Чтобы гарантировать сходимость классического метода условного градиента, необходимо правильно выбрать длину шага γ_k. Рассмотрим две стратегии выбора γ_k.

1. Последовательность $\{\gamma_k\}$ выбирается заранее:

$$\gamma_k = \frac{2}{k+1}, \quad \text{где } k = 1, 2, \ldots. \tag{2}$$

2. Наискорейший спуск:

$$\gamma_k = \underset{\gamma\in[0,\,1]}{\operatorname{argmin}}\, f(y_{k-1} + (1-\gamma)x_k). \tag{3}$$

Выделим класс функций с липшицевым градиентом.

Определение 1. Будем говорить, что функция f имеет непрерывный по Липшицу (или просто липшицев) градиент, если

$$\|\nabla f(x) - \nabla f(y)\|_* \leqslant L\|x - y\| \ \forall x, y \in X. \tag{4}$$

Пусть у нас есть выпуклая функция с липшицевым градиентом. Известно, что классическому методу условного градиента с длиной шага, выбранной по правилу (2) или (3), требуется

$$O\left(\frac{LD_X^2}{\varepsilon}\right) \tag{5}$$

итераций, чтобы достигнуть точности ε по функции, где $D_X = \max\limits_{x, y \in X} \|x - y\|$ [Frank, Wolfe, 1956]. Заметим, что, несмотря на то, что в методе условного градиента нам не нужно выбирать норму, в оценку сложности метода входят зависящие от нее константы $L = L_{\|\cdot\|}$, $D_X = D_{X, \|\cdot\|}$. Так как оценка (5) верна для произвольной нормы, то сложность данного метода можно оценить точнее:

$$O\left(\inf_{\|\cdot\|} \frac{L_{\|\cdot\|} D^2_{X, \|\cdot\|}}{\varepsilon}\right). \tag{6}$$

Размерность в методах условного градиента

Будем рассматривать различные методы условного градиента. Классический метод Франк – Вульфа является одним из них. Все эти методы итеративно используют линейный минимизационной оракул для решения задачи (1).

Определение 2. Для данного вектора $p \in \mathbb{R}^n$ линейный минимизационный оракул ($\mathrm{LMO}(p)$) вычисляет решение следующей задачи:

$$\mathrm{LMO}(p) \in \operatorname*{Argmin}_{x \in X} \langle p, x \rangle. \tag{7}$$

В общем виде алгоритм методов условного градиента можно записать следующим образом [Lan, 2013].

Алгоритм 2. Общий вид методов условного градиента

Дано: $x_0 \in X$ — начальная точка.
for $k = 1, 2, \ldots$ **do**
  Определяем вектор $p_k \in \mathbb{R}^n$.
  Вычисляем $x_k = \mathrm{LMO}(p_k)$.
  Возвращаем $y_k = \mathrm{Conv}\{x_0, \ldots, x_k\}$.
end for

Заметим, что в алгоритме 2 не указаны конкретное правило вычисления y_k и стратегия выбора p_k. Это делает алгоритм достаточно общим. Например, алгоритм 2 включает в себя, как частный случай, классический метод условного градиента (алгоритм 1).

Рассмотрим задачу (1) с функцией f с липшицевым градиентом. Для методов условного градиента известна следующая нижняя оценка [Lan, 2013]. Число итераций алгоритма 2 для решения этой задачи с точностью ε по функции не превосходит

$$\left\lceil \min\left\{\frac{n}{2}, \frac{LD_X^2}{4\varepsilon}\right\}\right\rceil - 1. \tag{8}$$

Заметим, что для сильно выпуклых функций нижняя оценка остается такой же (см. текст после теоремы 1 в [Lan, 2013]).

Пусть в сильно выпуклом случае сложность некоторого метода условного градиента $\mathcal{M}$ (алгоритм 2) есть $O\left(\max\left(\ln^m \frac{1}{\varepsilon},\, 1\right)\right)$, где $m = 1, 2, \ldots$. Например, при $m = 1$ скорость сходимости метода является линейной, а при $m \geqslant 2$ — сублинейной. Ускорим метод условного градиента с помощью Каталиста (Catalyst) [Lin, Mairal, Harchaoui, 2018; Гасников, 2018].

Покажем, как получается ускорение. Введем функции

$$F_{\varkappa,x}(y) = f(y) + \frac{\varkappa}{2}\|y - x\|_2^2,$$
$$f_\varkappa(x) = \min_{y\in X} F_{\varkappa,x}(y) = F_{\varkappa,x}(y_\varkappa(x)).$$

Меняем исходную задачу (1) на следующую:

$$f_\varkappa(x) \to \min_{x\in X}, \tag{9}$$

где $\varkappa$ мы можем выбирать сами.

Конструкция Каталист состоит из двух вложенных циклов. Рассмотрим метод Монтейро – Свайтера, использующийся в Каталисте для ускорения методов [Гасников, 2018; Monteiro, Svaiter, 2013; Ivanova et al., 2019]. В данном случае удобно выбрать $\varkappa = L$.

Алгоритм 3. Метод Монтейро – Свайтера

Дано:

Инициализация: z_0, y_0, $A_0 = 0$

while критерий остановки не выполнен **do**

Вычисляем

$$a_{k+1} = \frac{\frac{1}{\varkappa} + \sqrt{\frac{1}{\varkappa^2} + \frac{4A_k}{\varkappa}}}{2}, \quad A_{k+1} = A_k + a_{k+1}$$

Вычисляем

$$x_{k+1} = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} z_k$$

Подбираем y_{k+1} так, чтобы выполнялось условие Монтейро – Свайтера

$$\left\|\nabla F_{\varkappa,x_{k+1}}(y_{k+1})\right\|_2 \leqslant \frac{\varkappa}{2}\left\|y_{k+1} - x_{k+1}\right\|_2 \tag{10}$$

Вычисляем

$$z_{k+1} = z_k - a_{k+1}\nabla f(y_{k+1})$$

end while

Условие Монтейро – Свайтера (10) позволяет вместо точного решения $y_\varkappa(x)$ вспомогательной задачи (9) искать неточное решение задачи [Гасников, 2018; Lin, Mairal, Harchaoui, 2018] с точностью ε по $F_{\varkappa,x_{k+1}}(x)$:

$$y_{k+1} \approx \operatorname*{argmin}_{x\in X} F_{\varkappa,x_{k+1}}(x). \tag{11}$$

Таким образом, на каждом шаге внутреннего цикла методом $\mathcal{M}$ решается задача (11), чтобы найти y_{k+1}, удовлетворяющий (10). Сложность решения этой задачи с точностью ε алгоритмом 2 равна $O\left(\max\left(\ln^m \frac{1}{\varepsilon},\, 1\right)\right)$.

Во внешней итерации конструкции Каталист используется алгоритм 3. Его сложность равна $O\left(\sqrt{\frac{LR^2}{\varepsilon}}\right)$ [Monteiro, Svaiter, 2013; Гасников, 2018]. Тогда итоговое время работы метода окажется следующим:

$$O\left(\sqrt{\frac{LR^2}{\varepsilon}}\max\left(\ln^m\frac{1}{\varepsilon},\ 1\right)\right).$$

Сравним его с нижней оценкой (8)

$$\sqrt{\frac{LR^2}{\varepsilon}}\max\left(\ln\frac{1}{\varepsilon}, =, 1\right) \geqslant \min\left\{\frac{n}{2},\ \frac{LR^2}{4\varepsilon}\right\}.$$

Предположим, что при данном n выбирается ε так, что $\frac{LR^2}{2\varepsilon} \approx n$. Пусть C_n — константа в оценке сложности метода, то есть метод сходится как $C_n \ln^m \frac{1}{\varepsilon} = O\left(\ln^m \frac{1}{\varepsilon}\right)$. Получаем

$$C_n \geqslant \frac{\sqrt{\frac{LR^2}{\varepsilon}}}{\ln^m \frac{LR^2}{\varepsilon}} = \frac{\sqrt{n}}{\ln^m n} = \widetilde{\Omega}\left(\sqrt{n}\right).$$

Таким образом, мы доказали следующую теорему.

Теорема 1. *Пусть у оптимизационного метода $\mathcal{M}$, имеющего структуру алгоритма* 2, *сложность в сильно выпуклом случае равна $C_n\left(\max\left(\ln^m \frac{1}{\varepsilon},\ 1\right)\right)$, где $m = 1, 2, \ldots$, а C_n — число, зависящее от n. Тогда $C_n = \widetilde{\Omega}\left(\sqrt{n}\right)$.*

Получается, что в оценке скорости сходимости методов типа условного градиента в сильно выпуклом случае нельзя избавиться от размерности.

Методы условного градиента для решения задачи квадратичной оптимизации на кубе

Из результатов предыдущего пункта следует, что методы условного градиента (алгоритм 2) будут неэффективны для решения многих сильно выпуклых оптимизационных задач, так как в оценке скорости сходимости таких алгоритмов нельзя избавиться от размерности. Ланом был предложен метод рестартов условного градиента (Shrinking Conditional Gradient) [Lan, 2013] для сильно выпуклых функций. В нем вместо линейного минимизационного оракула (определение 2) используется линейный оракул, способный решать оптимизационные задачи следующего вида:

$$\min\{\langle p,\ x\rangle\colon\ x \in X,\ \|x - x_0\| \leqslant R\}. \tag{12}$$

Стратегия выбора длины шага γ_k в алгоритме 4 такая же, как и в алгоритме 1, и задается формулами (2), (3). Рестартам условного градиента с такими правилами выбора шага требуется не более чем

$$\frac{8L}{\mu}\left\lceil \max\left(\frac{\mu D_X^2}{\varepsilon},\ 1\right)\right\rceil \tag{14}$$

итераций, чтобы достичь точности ε по функции [Lan, 2013].

Рассмотрим задачу квадратичной оптимизации на кубе (∞-шаре):

$$f(x) = \frac{1}{2}\|Ax\|_2^2 \to \min_{\|x\|_\infty \leqslant 1}. \tag{15}$$

Алгоритм 4. Рестарты условного градиента (Shrinking Conditional Gradient)

1: **Дано:** $p_0 \in X$ — начальная точка, $R_0 = D_X$.
2: **for** $t = 1, 2, \ldots$ **do**
3: Положим $y_0 = p_{t-1}$
4: **for** $k = 1, \ldots, \frac{8L}{\mu}$ **do**
5: Вычисляем x_k — решение задачи:

$$\underset{x \in X_{t-1}}{\mathrm{Argmin}} \langle f'(y_{k-1}), x \rangle, \quad \text{где } X_{t-1} = \{x \in X : \|x - p_{t-1}\| \leqslant R_{t-1}\}. \tag{13}$$

6: Положим $y_k = (1 - \gamma_k) y_{k-1} + \gamma_k x_k$ для некоторого $\gamma_k \in [0, 1]$.
7: **end for**
8: Положим $p_t = y_k$ и $R_t = \frac{R_{t-1}}{\sqrt{2}}$
9: **end for**

При этом мы считаем, что в каждом столбце и каждой строке матрицы A не более $s \ll n$ элементов отлично от нуля (A — разрежена), то есть всего в A может быть sn ненулевых элементов. Кроме того, A удовлетворяет условию $\mu I \leqslant A$, то есть задача (15) сильно выпукла.

Для решения этой задачи будем пользоваться алгоритмом 4. Сначала рассмотрим шаги внутреннего цикла этого метода (строки 5–6 алгоритма 4). Пусть в (13) $X_{t-1} = \{x : \|x\|_\infty \leqslant 1\}$. Решаем задачу

$$\underset{\|x\|_\infty \leqslant 1}{\mathrm{Argmin}} \langle f'(y_k), x \rangle. \tag{16}$$

Так как это задача нахождения минимума линейной формы на кубе, то ее решение — вершина этого куба. Нам известен градиент с предыдущего шага:

$$\nabla f(y_{k-1}) = \left(\frac{\partial f(y_{k-1})}{\partial x^1}, \ldots, \frac{\partial f(y_{k-1})}{\partial x^n} \right)^T.$$

Тогда решением задачи (16) является вектор

$$x_k = -\left(\mathrm{sign}\left(\frac{\partial f(y_{k-1})}{\partial x^1} \right), \ldots, \mathrm{sign}\left(\frac{\partial f(y_{k-1})}{\partial x^n} \right) \right)^T.$$

Сложность нахождения этого вектора есть $O(n)$.

Пересчитаем y:

$$y_k = (1 - \gamma_k) y_{k-1} + \gamma_k x_k,$$

и градиент:

$$\nabla f(y_k) = A^T A y_k = (1 - \gamma_k) A^T A y_{k-1} + \gamma_k A^T A x_k.$$

Сложность пересчета градиента — $O(s^2 n)$, так как сначала надо умножить разряженную матрицу A на n-мерный вектор x_k, а потом еще домножить на A^T.

Пусть мы вышли из внутреннего цикла алгоритма 4. Чтобы решить задачу (13), необходимо найти X_t — пересечение исходного множества X и шара в ∞-норме с центром в p_t и радиусом R_t. Опишем процедуру его получения.

На каждой итерации внешнего цикла алгоритма 4 мы будем делать препроцессинг: масштабировать систему координат и переносить ее центр, чтобы в пересечении получался шар $B_1^\infty(0)$. Так всегда можно сделать, потому что мы пересекаем n-мерные осепараллельные прямоугольники.

Введем обозначение:

$$B_R^\infty(c) \overset{\mathrm{def}}{=} \{x\colon \|x - c\|_\infty \leqslant R\}.$$

После масштабирования и изменения центров системы координат на предыдущих итерациях X имеет вид осепараллельного прямоугольника. Пусть его центр находится в точке c_t и задан вектор расстояний от центра до граней a_t (или координаты одной вершины, по которым можно вычислить эти расстояния). Проверим, лежит ли $B_{R_t}^\infty(p_t)$ в X. Это можно сделать за $O(n)$, проверив принадлежность отрезка $[p_t^i - R_t,\ p_t^i + R_t]$ отрезку $[c_t^i - a_t^i,\ c_t^i + a_t^i]$ для всех $i \in \{1, 2, \ldots, n\}$.

Пусть оказалось, что $B_{R_t}^\infty(p_t) \subseteq X$. Перенесем p_t в нуль. Обновим значение $y_0 = p_t - p_t = 0$. Сделаем масштабирование системы координат, чтобы привести шар $B_{R_t}^\infty(p_t) \subseteq X$ к единичному. Осталось пересчитать множество X в соответствии с произведенными преобразованиями:

$$c_{t+1} = \frac{c_t - p_t}{R_t}, \quad a_{t+1} = \frac{a_t}{R}.$$

В другом случае $B_{R_t}^\infty(p_t) \nsubseteq X$. В этом случае для всех $i \in [1, 2, \ldots, n]$ ищем пересечение $[p_t^i - R_t,\ p_t^i + R_t] \cap [c_t^i - a_t^i,\ c_t^i + a_t^i] = [\min^i, \max^i]$ и центры m_t^i этих пересечений. Вектор m_t — центр осепараллельного параллелепипеда, образованного пересечением $B_{R_t}^\infty(p_t)$ и X. Перенесем m_t в ноль и отмасштабируем по каждой из осей, чтобы сделать X_t единичным кубом. Следовательно, координаты любой точки $x \in \mathbb{R}^n$ преобразуются следующим образом:

$$x^i := \frac{2 \cdot (x^i - m_t^i)}{\max^i - \min^i}. \tag{17}$$

Начальная точка внутреннего цикла y_0 и центр c_t исходного множества X преобразуются в соответствии с этим правилом. Сложность этих вычислений есть $O(n)$.

Таким образом, мы свели задачу (13) к (16). Процедуру решения задачи (16) мы описали выше.

В итоге сложность шагов внутреннего цикла алгоритма 2 равна $O(sn)$, сложность препроцессинга — $O(n)$. Получаем следующую оценку сложности предложенного метода:

$$\frac{8L}{\mu} O(n) + \frac{8L}{\mu}\left(s^2 n \log \frac{\mu R_0}{\varepsilon}\right), \tag{18}$$

где $R_0 = 2$ — размер шара $B_1^\infty(0)$ в ∞-норме, а L — константа Липшица градиента в ∞-норме.

Сравним эту оценку с оценкой градиентного метода [Allen-Zhu, Orecchia, 2014]:

$$\frac{L}{\mu} O\left(s^2 n \log \frac{L_1 R_0^2}{\varepsilon}\right), \tag{19}$$

где $R_0^2 = O\left(\sqrt{n}\right)$.

Получается, что предложенный нами метод решения задачи (15) работает не сильно лучше градиентного метода. Заметим, что не получилось улучшения оценки, как в статье [Аникин и др., 2015], где методом Франк – Вульфа (алгоритм 1) решали задачу поиска вектора RageRank в выпуклом случае на единичном симплексе. Дело в том, что если задача поставлена на симплексе, то сложность пересчета градиента, при хранении его компонент в бинарной куче, занимает $O(s^2 \log n)$, так как необходимо обновлять всего одну компоненту градиента на каждой итерации. В случае ∞-шара надо обновлять все n компонент. Соответственно, сложность этой операции выше ($O(s^2 n)$).

Удобство постановки задачи на ∞-шаре заключается в том, что на каждой итерации сохраняется геометрия задачи. Пересечение ∞-шаров, на котором идет минимизация в задаче (13), есть многомерный осепараллельный прямоугольник. Стоит отметить, что пока неясно, как использовать алгоритм 4 для решения задачи PageRank на симплексе или 1-шаре. Пересечение 1-шара и шара в произвольной норме, или симплекса и шара, может оказаться сложным выпуклым многогранником. Непонятно, как найти этот многогранник. То есть уметь, например, вычислять все его вершины (в какой-то из них и будет минимум линейного функционала в задаче (13)). Несмотря на то что авторы статьи [Lan, 2013] указывают в ней, что сложность решения задачи (12) сравнима со сложностью решения задачи (7), пример про задачу PageRank на симплексе или 1-шаре показывает противное.

С другой стороны, в алгоритме 4 в задаче (13) не обязательно брать пересечение исходного множества с шаром, а можно брать некоторое множество из исходного, которое содержит в себе пересечение. Это повлияет только на константу в оценке скорости сходимости метода. Таким образом, на каждой итерации метода можно всегда решать задачу (13) на одном и том же множестве, например 1-шаре. Но и в этом случае непонятно, как именно найти множество, содержащее X_t.

Автор выражает благодарность Гасникову Александру Владимировичу за полезные обсуждения и идеи.

Список литературы (References)

Аникин А. С., Гасников А. В., Горнов А. Ю., Камзолов Д. И., Максимов Ю. В., Нестеров Ю. Е. Эффективные численные методы решения задачи PageRank для дважды разреженных матриц // Научные труды МФТИ. — 2015. — Т. 7, № 4. — С. 74–94.

Anikin A. S., Gasnikov A. V., Gornov A. Yu., Kamzolov D. I., Maximov Yu. V., Nesterov Yu. E. Effektivnye chislennye metody resheniya zadachi PageRank dlya dvazhdy razrezhennykh matrits [Efficient numerical methods for solving PageRank problem for doubly sparse matrices] // Nauchnye trudy MFTI. — 2015. — Vol. 7, No. 4. — P. 74–94 (in Russian).

Гасников А. В. Современные численные методы оптимизации. Метод универсального градиентного спуска. — М.: МФТИ, 2018.

Gasnikov A. V. Sovremennye chislennye metody optimizatsii. Metod universal'nogo gradientnogo spuska [Modern numerical optimization methods. The universal gradient descent method]. — M.: MFTI, 2018 (in Russian).

Гасников А. В. Эффективные численные методы поиска равновесий в больших транспортных сетях // Диссертация на соискание степени д. ф.-м. н. по специальности 05.13.18 – Математическое моделирование, численные методы, комплексы программ. — М.: МФТИ, 2016. — 487 с.

Gasnikov A. V. Effektivnye chislennye metody poiska ravnovesii v bol'shikh transportnykh setyakh [Efficient numerical methods for searching equillibriums in large transport networks]. — M.: MFTI, 2016. — 487 p. (in Russian).

Горнов А. Ю. Вычислительные технологии решения задач оптимального управления. — Новосибирск: Наука, 2009. — 277 с.

Gornov A. Yu. Vychislitel'nye tekhnologii resheniya zadach optimal'nogo upravleniya [Computational technologies for solving optimal control problems]. — Novosibirsk: Nauka, 2009. — 277 p. (in Russian).

Левитин Е. С., Поляк Б. Т. Методы минимизации при наличии ограничений // Журнал вычислительной математики и математической физики. — 1966. — Т. 6, № 5. — С. 787–823.

Levitin E. S., Polyak B. T. Constrained minimization methods // U.S.S.R. Comput. Math. Math. Phys. — 1966. — Vol. 6, No. 5. — P. 1–50. — DOI: 10.1016/0041-5553(66)90114-5 (Original Russian Paper: *Levitin E. S., Polyak B. T.* Metody minimizatsii pri nalichii ogranicheniy // Zhurnal vychislitel'noi matematiki i matematicheskoy fiziki. — 1966. — Vol. 6, No. 5. — P. 787–823.)

Allen-Zhu Z., Orecchia L. Linear coupling: An ultimate unification of gradient and mirror descent // arXiv preprint. — 2014. — https://arxiv.org/pdf/1407.1537

Brin S., Page L. The anatomy of a large-scale hypertextual Web search engine // Comput. Network ISDN Syst. — 1998. — Vol. 30 (1–7). — P. 107–117. — DOI: 10.1016/S0169-7552(98)00110-X

Dvurechensky P., Shtern S., Staudigl M., Ostroukhov P., Safin K. Self-concordant analysis of Frank–Wolfe algorithms // arXiv preprint. — 2020. — https://arxiv.org/pdf/2002.04320

Frank M., Wolfe P. An algorithm for quadratic programming // Naval research logistics quarterly. — 1956. — Vol. 3, No. 1–2. — P. 95–110. — DOI: 10.1002/nav.3800030109

Garber D., Hazan E. A Linearly Convergent Conditional Gradient Algorithm with Applications to Online and Stochastic Optimization // SIAM Journal on Optimization. — 2013. — Vol. 26. — DOI: 10.1137/14098536

Ivanova A., Pasechnyuk D., Grishchenko D., Shulgin E., Gasnikov A. Adaptive Catalyst for Smooth Convex Optimization // arXiv preprint. — 2019. — https://arxiv.org/pdf/1911.11271

Jaggi M. Revisiting Frank – Wolfe: Projection-Free Sparse Convex Optimization // Proceedings of the 30th International Conference on Machine Learning. — 2013. — Vol. 28, No. 1. — P. 427–435.

Kerdreux T., d'Aspremont A., Pokutta S. Restarting Frank – Wolfe // Proceedings of Machine Learning Research. — 2019. — Vol. 89. — P. 1275–1283.

Lacoste-Juien S., Jaggi M. On the global linear convergence of Frank – Wolfe optimization variants // Advances in Neural Information Processing Systems. — 2015. — Vol. 28. — P. 469–504.

Lan G. The Complexity of Large-scale Convex Programming under a Linear Optimization Oracle // arXiv preprint. — 2013. — https://arxiv.org/pdf/1309.5550

Lin H., Mairal J., Harchaoui Z. Catalyst Acceleration for First-order Convex Optimization: from Theory to Practice // Journal of Machine Learning Research. — 2018. — Vol. 18, No. 212. — P. 1–54.

Monteiro R., Svaiter B. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods // SIAM Journal on Optimization. — 2013. — Vol. 23, No. 2. — P. 1092–1125.

Carderera A., Diakonikolas J., Lin C. Y., Pokutta S. Parameter-free locally accelerated conditional gradients // International Conference on Machine Learning. — PMLR, 2021. — P. 1283–1293.