ПРОГРАММИРОВАНИЕ, 2023, № 6, с. 27–35

АНАЛИЗ ДАННЫХ

УДК 519.85

ДЕЦЕНТРАЛИЗОВАННЫЙ МЕТОД УСЛОВНОГО ГРАДИЕНТА НА ПЕРЕМЕННЫХ ВО ВРЕМЕНИ ГРАФАХ

© 2023 г. Р. А. Ведерников[a,*], А. В. Рогозин[a,] (ORCID: 0000-0003-3435-268), А. В. Гасников[b,c,***] (ORCID: 0000-0002-7386-039X)**

[a] *Московский физико-технический институт*
141701 г. Долгопрудный, Институтский пер., д. 9, Россия

[b] *Институт проблем передачи информации РАН им. А.А. Харкевича*
127051 Москва, Большой Каретный пер., д. 19, стр. 1, Россия

[c] *Кавказский математический центр Адыгейского государственного университета*
385016 Республика Адыгея, г. Майкоп, ул. Первомайская, д. 208, Россия

**E-mail: vedernikov.ra@phystech.edu*

***E-mail: aleksandr.rogozin@phystech.edu*

****E-mail: gasnikov@yandex.ru*

Поступила в редакцию 13.06.2023 г.
После доработки 14.07.2023 г.
Принята к публикации 20.07.2023 г.

В данной работе рассматривается обобщение децентрализованного алгоритма Франк-Вульфа на переменные во времени сети, исследуются свойства сходимости алгоритма и проводятся соответственные численные эксперименты. Меняющаяся сеть моделируется как детерминированная или стохастическая последовательность графов.

DOI: 10.31857/S0132347423060080, **EDN:** FDENUK

1. ВВЕДЕНИЕ

Алгоритм Франк–Вульфа [1], (также известен как метод условного градиента или метод Левитина–Поляка [2]) – итеративный алгоритм оптимизации, который часто используется для решения задач выпуклой оптимизации. Он был представлен Маргаритой Франк и Филиппом Вульфом в 1956 году.

Алгоритм 1. Классический метод условного градиента

Require: Количество итераций m, начальная точка $x_0 \in Q$.

1: **for** $t = 0,1,\ldots,m-1$ **do**

2: $\alpha_t = \dfrac{2}{t+1}$

3: $s_t = \arg\min_{x\in Q}\{\nabla f(x_t)^\top x\}$

4: $x_{t+1} = (1-\alpha_t)x_t + \alpha_t s_t$

5: **end for**

Ensure: x_m

Основная идея алгоритма Франк–Вульфа заключается в следующем:

Алгоритм инициализируется в пределах допустимого множества D. После этого начинаются итерации алгоритма: на каждой итерации мы приближаем целевую функцию линейной функцией в окрестности текущей точки, и ищем точку из допустимого множества, проекция которой на направление антиградиента будет максимальной. Эта точка задает направление шага алгоритма, которое, вообще говоря, может не совпадать с направлением антиградиента, что отличает его от градиентного спуска.

После выбора направления движения, есть два основных способа задать величину шага. Первый – выбрать шаг заранее и задать функцией от номера итерации:

$$\gamma_t = \frac{2}{t+2}, \tag{1.1}$$

Второй способ – техника short step rule, которая заключается в решении задачи минимизации функции на выбранном направлении по допустимому множеству на каждой итерации:

$$\gamma_t = \arg\min_{\gamma\geqslant 0} f(x_t + \gamma(s_t - x_t)). \tag{1.2}$$

Подзадача линейной минимизации часто проще, чем исходная задача. Алгоритм может быть особенно полезен, когда допустимая область яв-

ляется компактным и выпуклым множеством в пространстве большой размерности.

В работе рассматривается приложение алгоритма Франк–Вульфа к решению задач на сетях, в силу особенностей топологии не имеющих общего распределяющего центра и требующих применения децентрализованного алгоритма.

2. ДЕЦЕНТРАЛИЗОВАННАЯ ОПТИМИЗАЦИЯ

2.1. Постановка задачи

Рассмотрим произвольную систему из N узлов. Узлы могут обмениваться информацией через переменную во времени сеть, каждый узел связан с некоторыми другими, может передавать и получать информацию только от них. Таким образом, систему можно представить последовательностью неориентированных графов $G^t = (V, E^t)$, причем в ней нет главного узла, который мог бы аггрегировать информацию. Будем требовать, чтобы на каждой итерации граф системы оставался связным.

Будем рассматривать задачу минимизации суммы функции:

$$\min_{x \in D} f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x). \tag{2.1}$$

Эта задача относится к задачам децентрализованной оптимизации. Каждый i-й узел хранит состояние x, умеет вычислять свою функцию $f_i(x)$, а также ее градиент $\nabla f(x)$ в этой точке.

Задачи такого типа имеют множество приложений в областях, где ограничена или невозможна аггрегация информации из-за ограничений безопасности, архитектуры сети или размеров данных, например, в распределенном машинном обучении [3, 4], системах контроля мощностей [5, 6], контроле и управлении техникой [7].

2.2. Коммуникационная матрица

Важную роль в алгоритмах децентрализованной оптимизации играет процесс консенсуса, который реализует обмен информации между узлами.

Коммуникационная матрица [8] – это матрица, используемая в этом процессе, где каждый элемент представляет собой силу связи или вес между двумя узлами. Коммуникационная матрица является важной частью алгоритма консенсуса, который помогает всем узлам достичь согласия относительно оптимального решения.

В процессе децентрализованной оптимизации алгоритм консенсуса работает следующим образом:

- Каждый узел начинает с начальной оценки решения.
- Каждый узел сообщает свою оценку своим соседям.
- Каждый узел обновляет свою оценку на основе полученной от своих соседей информации, взвешенной согласно коммуникационной матрице.

Этот процесс повторяется до тех пор, пока все узлы не достигнут консенсуса, то есть их оценки не сойдутся к одному и тому же значению. Веса в коммуникационной матрице могут быть скорректированы в соответствии с потребностями системы, например, чтобы дать больший вес оценкам узлов, которые известны своей большей точностью или надежностью.

На последовательность коммуникационных матриц накладываются следующие условия:

Предположение 1. *Для каждого* $t = 0,1,\ldots$ *выполняется*

1. (*Согласованность* с *графом*) $[W^t]_{ij} = 0$ *if* $(i, j) \notin E^t$ *и* $i \neq j$.

2. (*Дважды стохастичность*) $W^t \mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top W^t = \mathbf{1}^\top$.

3. (*Свойство сжатия*) *Найдется такое* $\lambda < 1$, *что для любого* $t = 0,1,\ldots$ *выполняется*

$$\left\| \left(W^t - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \right) x \right\| \leqslant \lambda \|x\|.$$

Построение коммуникационной матрицы такой, чтобы она обладала свойством сжатия, может быть неочевидным. Приведем достаточные условия для выполнения данного свойства. Пусть для любого $t = 0,1,\ldots$ выполняется:

1. $W^t \mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top W^t = \mathbf{1}^\top$.

2. Для любого $i = 1,\ldots,N$ выполняется $\left[W^t\right]_{ii} > 0$.

3. Если $(i, j) \in E^t$, то $\left[W^t\right]_{ij} > 0$, иначе $\left[W^t\right]_{ij} = 0$.

4. Существует $\theta > 0$, такое что если $\left[W^t\right]_{ij} > 0$, то $\left[W^t\right]_{ij} \geqslant \theta$.

В этой работе мы будем использовать способ выбора весов Metropolis Weights, который имеет следующий вид:

$$\left[W^t\right]_{ij} = \begin{cases} \dfrac{1}{\max(\deg(i),\deg(j)) + 1}, & (i, j) \in E^t, \\ 0, & (i, j) \notin E^t, \\ 1 - \sum\limits_{i \neq j} \left[W^t\right]_{ij}, & i = j. \end{cases} \tag{2.2}$$

2.3. Построение алгоритма

Децентрализованный алгоритм Франк–Вульфа строится из его классической версии [9]. Пусть $t \in \mathbb{N}$ – номер итерации, а начальная точка $\theta_0 \in D$ взята из допустимого множества. Напомним, что $F(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x^i)$ (где x – матрица, строками которой являются $(x^i)^\top$), тогда централизованный алгоритм действует следующим образом:

$$v_t \in \arg\min_{v \in D^N} \langle \nabla F(x_t), v \rangle, \tag{2.3}$$

$$x_t = x_{t-1} + \gamma_{t-1}(v_{t-1} - x_{t-1}), \tag{2.4}$$

где $\gamma_{t-1} \in (0,1]$ – заданная величина шага алгоритма. Заметим, что x_t – выпуклая комбинация x_{t-1} and v_{t-1}, которые лежат в допустимом множестве, поэтому также принадлежит допустимому множеству. Когда шаг алгоритма задается как $\gamma_t = 2/(t+2)$, известна оценка скорости сходимости $O(1/t)$, если F является L-гладкой и выпуклой функцией. Следующим этапом будет децентрализация алгоритма. Для этого необходимо заменить централизованные значения функции и градиента на их локальные приближения. Во-первых, определим среднее координат точек:

$$\overline{x}_t = \frac{1}{N}\sum_{i=1}^{N} x_t^i \tag{2.5}$$

и среднее градиентов в координатах узлов:

$$\overline{\nabla_t F} = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\overline{x}_t^i). \tag{2.6}$$

Эти величины понадобятся нам, чтобы оценить скорость сходимости алгоритма. Во-вторых, определим локальные аппроксимации этих величин, доступные для вычисления в каждом узле.

Вычисление локальной аппроксимации точки консенсуса $\overline{x}_t^i$ называется шагом консенсуса (consensus step):

$$\overline{x}_t^i = \sum_{j=1}^{N} W_{ij}^t \cdot x_t^j. \tag{2.7}$$

Здесь реализуется связь соседних узлов, причем информация из несвязанных между собой узлов этими узлами игнорируется, т.к. $W_{ij}^t = 0, (i,j) \notin E^t$.

Вычисление локальной аппроксимации градиента $\overline{\nabla_t^i F}$ выполняется по другой схеме. Для этого сначала определим вспомогательный градиент:

$$\nabla_t^i F = \overline{\nabla_{t-1}^i F} + \nabla f_i(\overline{x}_t^i) - \nabla f_i(\overline{x}_{t-1}^i). \tag{2.8}$$

После того, как каждый узел посчитает свой вспомогательный градиент, выполняется шаг агрегации (aggregate step):

$$\overline{\nabla_t^i F} = \sum_{j=1}^{N} W_{ij}^t \cdot \nabla_t^j F. \tag{2.9}$$

В следующей главе приведено описание децентрализованного алгоритма Франка–Вульфа.

3. ТЕОРЕТИЧЕСКИЕ ОЦЕНКИ

3.1. Верхняя оценка скорости сходимости

Как было указано выше, нам нужны величины $\overline{x}_t$, $\overline{\nabla_t F}$, чтобы следить, насколько результаты consensus step и aggregation step отличаются от среднего. Введем несколько предположений, чтобы сделать оценку скорости сходимости алгоритма:

Алгоритм 2. Децентрализованный метод условного градиента

Require: Начальные точки $x_0^i \in D$ $(i = 1,\ldots,N)$, целевая функция F, константа гладкости L.

1: **for** $t = 0,1,\ldots$ **do**

2: Консенсусный шаг:

$$\overline{x}_t^i \leftarrow \sum_{j=1}^{N} W_{ij}^t \cdot x_t^j, \quad \forall i \in V$$

3: Шаг агрегации:

$$\overline{\nabla_t^i F} \leftarrow \sum_{j=1}^{N} W_{ij}^t \cdot \nabla_t^j F, \quad \forall i \in V$$

4: $v_t^i \leftarrow \arg\min_{v \in D} \langle \overline{\nabla_t^i F}, v \rangle$

5: $\gamma_t \leftarrow 2/(t+2)$

6: $x_{t+1}^i \leftarrow \overline{x}_t^i + \gamma_t(v_t^i - \overline{x}_t^i)$

7: **end for**

Предположение 2. *Для каждого $i = 1,\ldots,N$ функция f_i является выпуклой L-гладкой, т.е. для всяких $x, y \in D$ выполняется*

$$0 \leqslant f_i(y) - f_i(x) - \langle \nabla f_i(x), y - x \rangle \leqslant \frac{L}{2}\|y - x\|_2^2.$$

Предположение 3. *Пусть существует $(\{\Delta p_t\}_{t \geqslant 1})$, $\forall t \geqslant 1$ неотрицательная последовательность такая, что $\Delta p_t \to 0$ и*

$$\max_{i \in [N]} \|\overline{x}_t^i - \overline{x}_t\|_2 \leqslant \Delta p_t. \tag{3.1}$$

Предположение 4. *Пусть существует $\{\Delta d_t\}_{t \geqslant 1}$, $\forall t \geqslant 1$ неотрицательная последовательность такая, что $\Delta d_t \to 0$ и*

$$\max_{i \in [N]} \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2 \leqslant \Delta d_t. \tag{3.2}$$

Отметим, что для выполнения предыдущего предположения достаточно определять коммуникационную матрицу способом Metropolis Weights, описанном в (2.2). Отсюда следует [9] следующая оценка:

Теорема 1. *Пусть выполнены предположения* 1, 2, 3, 4, *размер шага равен* $\gamma_t = 2/(t+1)$, *а* C_p, C_g – *положительные константы*, *такие что* $\Delta p_t = C_p/t$, $\Delta d_t = C_g/t$. *Тогда*

$$F(\overline{x}_t) - F(\overline{x}^*) \leqslant \frac{8\overline{\rho}(C_g + LC_p) + 2L\overline{\rho}^2}{t+1} \qquad (3.3)$$

для любых $t \geqslant 1$, *где* $\overline{x}^*$ – *оптимальное решение задачи*.

Таким образом, при выполнении предположений получаем линейную оценку скорости сходимости алгоритма.

Сформулируем и докажем леммы, которые гарантируют выполнение вышеизложенных предположений. Леммы приведем в общем виде для произвольного параметра $\alpha \in (0,1]$, хотя нам понадобится только $\alpha = 1$.

Лемма 1. *Пусть* $t_0(\alpha)$ – *наименьшее положительное целое число такое*, *что*

$$\max_t \lambda_2(W^t) \leqslant \left(\frac{t_0(\alpha)}{t_0(\alpha)+1}\right)^{\alpha} \cdot \frac{1}{1+(t_0(\alpha))^{-\alpha}}. \qquad (3.4)$$

Зададим шаг $\gamma_t = 1/t^{\alpha}$ *в алгоритме Франк–Вульфа для* $\alpha \in (0,1]$, *тогда выполняется*:

$$\max_{i\in V} \|\overline{x}_t^i - \overline{x}_t\|_2 \leqslant \Delta p_t = C_p/t^{\alpha}, \quad \forall t \geqslant 1, \qquad C_p = (t_0(\alpha))^{\alpha} \cdot \sqrt{N}\rho. \qquad (3.5)$$

Доказательство. В доказательстве будем писать $t_0 = t_0(\alpha)$. Покажем, что

$$\sqrt{\sum_{i=1}^N \|\overline{x}_t^i - \overline{x}_t\|_2^2} \leqslant \frac{C_p}{t^{\alpha}}, \quad C_p = (t_0)^{\alpha} \cdot \sqrt{N}\rho. \qquad (3.6)$$

Заметим, что от $t = 1$ до $t = t_0$ неравенство выполняется, т.к. $\overline{x}_t^i$, $\overline{x}_t$ принадлежат допустимому множеству, и его диаметр ограничен ρ. Для шага индукции предположим, что неравенство выполняется для $t \geqslant t_0$. По определению,

$$x_{t+1}^i = (1 - t^{-\alpha})\overline{x}_t^i + t^{-\alpha} v_t^i.$$

Обозначим $\tilde{a}_t = \frac{1}{N}\sum_{j=1}^N a_t^i$. Так как для $\overline{x}_i = \sum_{j=1}^N W_{ij} \cdot x_j$ выполняется

$$\sqrt{\sum_{i=1}^N \|\overline{x}_i - \overline{x}\|^2} \leqslant |\lambda_2(W^t)| \cdot \sqrt{\sum_{i=1}^N \|x_i - \overline{x}\|^2},$$

то получаем

$$\sum_{i=1}^N \|\overline{x}_{t+1}^i - \overline{x}_{t+1}\|_2^2 \leqslant |\lambda_2(W^t)|^2 \times \times \sum_{j=1}^N \|(1 - t^{-\alpha})(\overline{x}_t^j - \overline{x}_t) + t^{-\alpha}(v_t^j - \tilde{v}_t)\|_2^2$$

Важно отметить, что показанное выше неравенство справедливо только для рассматриваемого шага t, так как на каждой итерации алгоритма может меняться W^t, а значит и $\lambda_2(W^t)$. С другой стороны, количество случайных графов на N вершинах конечно, а значит, $\forall t\ \lambda_2(W^t) \leqslant \lambda = \max_t \lambda_2(W^t)$.

В таком случае,

$$\sum_{i=1}^N \|\overline{x}_{t+1}^i - \overline{x}_{t+1}\|_2^2 \leqslant$$

$$\leqslant \lambda^2 \sum_{j=1}^N \|(1 - t^{-\alpha})(\overline{x}_t^j - \overline{x}_t) + t^{-\alpha}(v_t^j - \tilde{v}_t)\|_2^2 \leqslant$$

$$\leqslant \lambda^2 \sum_{j=1}^N \Big((1 - t^{-\alpha})^2 \|\overline{x}_t^j - \overline{x}_t\|_2^2 + \rho^2 t^{-2\alpha} + 2t^{-2\alpha}(1 - t^{-\alpha})^2 \rho \|\overline{x}_t^j - \overline{x}_t\|_2\Big) \leqslant$$

$$\leqslant \lambda^2 \sum_{j=1}^N \Big(\|\overline{x}_t^j - \overline{x}_t\|_2^2 + \rho^2 t^{-2\alpha} + 2\rho t^{-\alpha} \|\overline{x}_t^j - \overline{x}_t\|_2\Big) \leqslant$$

$$\overset{①}{\leqslant} \lambda^2 t^{-2\alpha}(C_p^2 + N\rho^2) + 2\rho t^{-\alpha}\sqrt{N}\sqrt{\sum_{j=1}^N \|\overline{x}_t^j - \overline{x}_t\|_2} \leqslant$$

$$\leqslant \lambda^2 t^{-2\alpha}(C_p + \sqrt{N}\rho) \leqslant \left(\lambda C_p \frac{(t_0)^{\alpha}+1}{(t_0)^{\alpha} \cdot t^{\alpha}}\right)^2.$$

Здесь в ① был использован тот факт, что для неотрицательных $c_1, \ldots, c_N \in \mathbb{R}$ выполняется

$$\sum_{j=1}^N c_j \leqslant \sqrt{N}\sqrt{\sum_{j=1}^N c_j^2}.$$

Из (3.4) получаем, что:

$$\lambda \cdot \frac{(t_0)^{\alpha}+1}{(t_0)^{\alpha} \cdot t^{\alpha}} \leqslant \frac{1}{(t+1)^{\alpha}}, \qquad (3.7)$$

шаг индукции выполняется, тогда

$$\sqrt{\sum_{i=1}^N \|\overline{x}_t^i - \overline{x}_t\|_2^2} \leqslant \frac{C_p}{t^{\alpha}}, \quad C_p = (t_0)^{\alpha} \cdot \sqrt{N}\rho, \qquad (3.8)$$

откуда следует доказываемое утверждение.

Доказанная лемма гарантирует выполнение условия на скорость сходимости последовательности точек, в следующей лемме рассмотрим скорость сходимости последовательности градиентов.

Напомним обозначения:

$$\nabla_t^i F = \overline{\nabla_{t-1}^i F} + \nabla f_i(\overline{x}_t^i) - \nabla f_i(\overline{x}_{t-1}^i),$$

$$\overline{\nabla_t^i F} = \sum_{j=1}^N W_{ij}^t \cdot \nabla_t^j F.$$

Лемма 2. *Зададим шаг* $\gamma_t = 1/t^\alpha$ *в алгоритме Франк–Вульфа для* $\alpha \in (0,1]$, *каждая из функций* f_i L *– гладкая, тогда выполняется*:

$$\max \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2 \leqslant \frac{C_g}{t^\alpha}, \qquad C_g = 2\sqrt{N}(t_0)^\alpha (2C_p + \overline{\rho})L. \tag{3.9}$$

Доказательство. Заметим, что от $t = 1$ до $t = t_0$ неравенство выполняется, что следует из ограниченности градиентов. Для шага индукции предположим, что неравенство выполняется для $t \geqslant t_0$.

Определим вспомогательную переменную $\delta f_{t+1}^i = \nabla f_i(\overline{x}_{t+1}^i) - \nabla f_i(\overline{x}_t^i)$. Тогда перепишем $\nabla_{t+1}^i F = \delta f_{t+1}^i + \nabla_t^i F$, а также $\overline{\nabla_{t+1}^i F} = \sum_{j=1}^N W_{ij} \nabla_{t+1}^j F$, получаем, оценив $\lambda_2(W^t)$ аналогично прошлому доказательству:

$$\sum_{i=1}^N \|\overline{\nabla_{t+1}^i F} - \overline{\nabla_{t+1} F}\|_2^2 \leqslant$$
$$\leqslant (\lambda_2(W^t))^2 \cdot \sum_{i=1}^N \|\overline{\nabla_t^i F} + \delta f_{t+1}^i - \overline{\nabla_{t+1} F}\|_2^2 \leqslant$$
$$\leqslant \lambda \cdot \sum_{i=1}^N \|\overline{\nabla_t^i F} + \delta f_{t+1}^i - \overline{\nabla_{t+1} F}\|_2^2.$$

Аналогично, определим $\delta F_{t+1} = \overline{\nabla_{t+1} F} - \overline{\nabla_t} F$, тогда правую часть (3.12) с помощью неравенства Коши–Буняковского–Шварца можно ограничить как

$$\sum_{i=1}^N \|\overline{\nabla_t^i F} + \delta f_{t+1}^i - \overline{\nabla_{t+1} F}\|_2^2 \leqslant$$
$$\leqslant \sum_{i=1}^N \Big(\|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2^2 + \|\delta f_{t+1}^i - \delta F_{t+1}\|_2^2 +$$
$$+ 2 \cdot \|\delta f_{t+1}^i - \delta F_{t+1}\|_2 \cdot \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2 \Big).$$

Кроме того, справедливо

$$\|\delta f_{t+1}^i\|_2 = \|\nabla f_i(\overline{x}_{t+1}^i) - \nabla f_i(\overline{x}_t^i)\|_2 \leqslant L\|\overline{x}_{t+1}^i - \overline{x}_t^i\|_2 \leqslant$$
$$\leqslant L \left\| \sum_{j=1}^N W_{ij}((x_{t+1}^j - \overline{x}_t^j) + (\overline{x}_t^j - \overline{x}_t^i)) \right\|_2 \leqslant$$
$$\leqslant L \sum_{j=1}^N W_{ij}(t^{-\alpha}\rho + 2C_p t^{-\alpha}) = (2C_p + \rho) L t^{-\alpha},$$

где последнее неравенство записано с помощью результата Леммы 5.

Используя неравенство треугольника, оценим еще одно слагаемое из (3.14):

$$\|\delta f_{t+1}^i - \delta F_{t+1}\|_2 = \left\| \left(1 - \frac{1}{N}\right)\delta_{t+1}^i + \frac{1}{N}\sum_{j\neq i} \delta_{t+1}^j \right\|_2 \leqslant$$
$$\leqslant \left(1 - \frac{1}{N}\right)\|\delta_{t+1}^i\|_2 + \frac{1}{N}\sum_{j\neq i}\|\delta_{t+1}^j\|_2 \leqslant$$
$$\leqslant 2\left(1 - \frac{1}{N}\right)(2C_p + \overline{\rho})\, L t^{-\alpha} \leqslant 2(2C_p + \overline{\rho})\, L t^{-\alpha}.$$

Итого, получаем окончательную оценку (3.14):

$$\sum_{i=1}^N \|\overline{\nabla_t^i F} + \delta f_{t+1}^i - \overline{\nabla_{t+1} F}\|_2^2 \leqslant$$
$$\leqslant t^{-2\alpha}(C_g^2 + 4N(2C_p + \rho)^2 L^2) +$$
$$+ t^{-\alpha} 4L(2C_p + \rho)\sqrt{N}\sqrt{\sum_{i=1}^N \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2^2} \leqslant$$
$$\leqslant t^{-2\alpha} \cdot \left(C_g + 2L\sqrt{N}(2C_p + \rho)\right)^2 \leqslant \left(\frac{(t_0)^\alpha + 1}{(t_0)^\alpha \cdot t^\alpha} \cdot C_g\right)^2.$$

Взяв корень из обеих частей неравенства, получаем:

$$\sqrt{\sum_{i=1}^N \|\overline{\nabla_{t+1}^i F} - \overline{\nabla_{t+1} F}\|_2^2} \leqslant \lambda\left(\frac{(t_0)^\alpha + 1}{(t_0)^\alpha \cdot t^\alpha} \cdot C_g\right). \tag{3.10}$$

И, с учетом (3.8), окончательно завершаем шаг индукции, откуда следует (3.10).

Таким образом, теорема 1 дает нам верхнюю оценку скорости сходимости алгоритма, а лемма 1 и лемма 2 гарантируют выполнение нужных предположений. Осталось заметить, что для выполнения условия (3.4) достаточно взять

$$t_0 = \left\lceil \frac{2}{1 - \lambda} \right\rceil.$$

Тогда, согласно определениям (3.5) и (3.9), получим

$$LC_p + C_g = O(NL\overline{\rho}^{-2}\chi^2).$$

Получаем окончательную скорость сходимости.

Следствие 1. *Для достижения точности* ε, *т.е. для выполнения условия*

$$\frac{1}{N}\sum_{i=1}^N f_i(x_N^i) - f(x^*) \leqslant \varepsilon, \tag{3.11a}$$

$$\max_{i\in[N]} \mathbb{E}\left\|\overline{x}_N^i - \overline{x}_N\right\|_2 \leqslant \varepsilon, \tag{3.11b}$$

необходимо

$$N = O\left(\frac{1}{(1-\lambda)^2}\frac{L\overline{\rho}^{-2}}{\varepsilon}\right)$$

итераций алгоритма 2.

Также заметим, что можно применить консенсусную процедуру, в которой коммуникационная матрица W^t заменится на последовательность матриц $W^{t+\tau-1}\dots W^t$, где $\tau = \lceil \chi \rceil$. Это позволит получить следующий результат.

Следствие 2. *Для достижения точности* ε (*см.* (3.11)) *с использованием консенсусной процедуры необходимо*

$$N_{comm} = O\left(\frac{1}{1-\lambda}\frac{L\rho^{-2}}{\varepsilon}\right)$$

коммуникационных шагов и

$$N_{orcl} = O\left(\frac{L\rho^{-2}}{\varepsilon}\right)$$

локальных вызовов линейного минимизационного оракула на каждом узле.

3.2. Случайная коммуникационная матрица

Можно провести аналогичные рассуждения, но в случае, когда матрица коммуникации имеет случайное распределение. Доказательство будет строиться на оценке отклонения от консенсуса $\overline{x}_t$ и $\overline{\nabla_t F}$. Так как матрица является стохастической, то во все невязки будут оцениваться по матожиданию. В данном разделе мы не приводим доказательства, так как они во многом повторяют часть с неслучайной матрицей. Введем соответствующие предположения.

Предположение 5. *На каждом шаге алгоритма матрица* W^t *является случайной и имеет распределение* $\mathcal{W}$. *Существует такое* $\lambda < 1$, *что для любого* $t = 0,1,\dots$ *выполняется*

1. $W^t\mathbf{1} = \mathbf{1}, \mathbf{1}^\top W^t = \mathbf{1}^\top$.
2. $\left[W^t\right]_{ij} = 0$ *если* $(i,j) \notin E^t$.
3. *Для любого* $x \in \mathbb{R}^n$ *выполняется*

$$\mathbb{E}\left\|\left(W^t - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)x\right\| \leqslant \lambda\|x\|.$$

Предположение 6. *Пусть существует* $(\{\Delta p_t\}_{t\geqslant 1})$, $\forall t \geqslant 1$ *неотрицательная последовательность такая, что* $\Delta p_t \to 0$, *тогда*

$$\max_{i\in[N]} \mathbb{E}\|\overline{x}_t^i - \overline{x}_t\|_2 \leqslant \Delta p_t \tag{3.12}$$

Предположение 7. *Пусть существует* $\{\Delta d_t\}_{t\geqslant 1}$, $\forall t \geqslant 1$ *неотрицательная последовательность такая, что* $\Delta d_t \to 0$, *тогда*

$$\max_{i\in[N]} \mathbb{E}\|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2 \leqslant \Delta d_t \tag{3.13}$$

Сформулируем основной результат для случайной матрицы коммуникаций.

Теорема 2. *Пусть выполняются предположения* 2, 5, 6, 7 *размер шага равен* $\gamma_t = 2/(t+1)$, *а также каждая из функций* f_i *выпуклая и* L*-гладкая. Пусть* C_p, C_g *– положительные константы, такие что* $\Delta p_t = C_p/t$, $\Delta d_t = C_g/t$. *Тогда*

$$\mathbb{E}F\left(\overline{x}_t\right) - F\left(\overline{x}^*\right) \leqslant \frac{8\overline{\rho}(C_g + LC_p) + 2L\overline{\rho}^2}{t+1} \tag{3.14}$$

для любых $t \geqslant 1$, *где* $\overline{x}^*$ *– оптимальное решение задачи.*

Таким образом, при выполнении предположений получаем линейную оценку скорости сходимости алгоритма.

Сформулируем леммы, которые гарантируют выполнение вышеизложенных предположений. Доказательство лемм аналогично доказательствам в разделе 3.1.

Лемма 3. *Пусть* t_0 *– наименьшее положительное целое число такое, что*

$$\lambda \leqslant \left(\frac{t_0(\alpha)}{t_0(\alpha)+1}\right)^\alpha \cdot \frac{1}{1+(t_0(\alpha))^{-\alpha}}. \tag{3.15}$$

Зададим шаг $\gamma_t = 1/t^\alpha$ *в алгоритме Франк–Вульфа для* $\alpha \in (0,1]$, *тогда выполняется*:

$$\max_{i\in V} \mathbb{E}\|\overline{x}_t^i - \overline{x}_t\|_2 \leqslant \Delta p_t = C_p/t^\alpha, \quad \forall t \geqslant 1 \tag{3.16}$$

$$C_p = (t_0(\alpha))^\alpha \cdot \sqrt{N\rho} \tag{3.17}$$

Лемма 4. *Зададим шаг* $\gamma_t = 1/t^\alpha$ *в алгоритме Франк–Вульфа для* $\alpha \in (0,1]$, *каждая из функций* f_i L*-гладкая*, *тогда выполняется*:

$$\max_{i\in V} \mathbb{E}\|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2 \leqslant \frac{C_g}{t^\alpha}, \tag{3.18}$$

$$C_g = 2\sqrt{N}(t_0)^\alpha(2C_p + \overline{\rho})L \tag{3.19}$$

Аналогично случаю с детерминированно изменяющейся матрицей, подытожим результаты.

Следствие 3. *Для достижения точности* ε, *т.е.*

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N f_i(x_N^i)\right] - f\left(x^*\right) \leqslant \varepsilon, \tag{3.20a}$$

$$\max_{i\in[N]} \mathbb{E}\|\overline{x}_N^i - \overline{x}_N\|_2 \leqslant \sqrt{\varepsilon}, \tag{3.20b}$$

необходимо

$$N = O\left(\frac{1}{(1-\lambda)^2}\frac{L\rho^{-2}}{\varepsilon}\right)$$

итераций. При использовании консенсусной процедуры необходимо

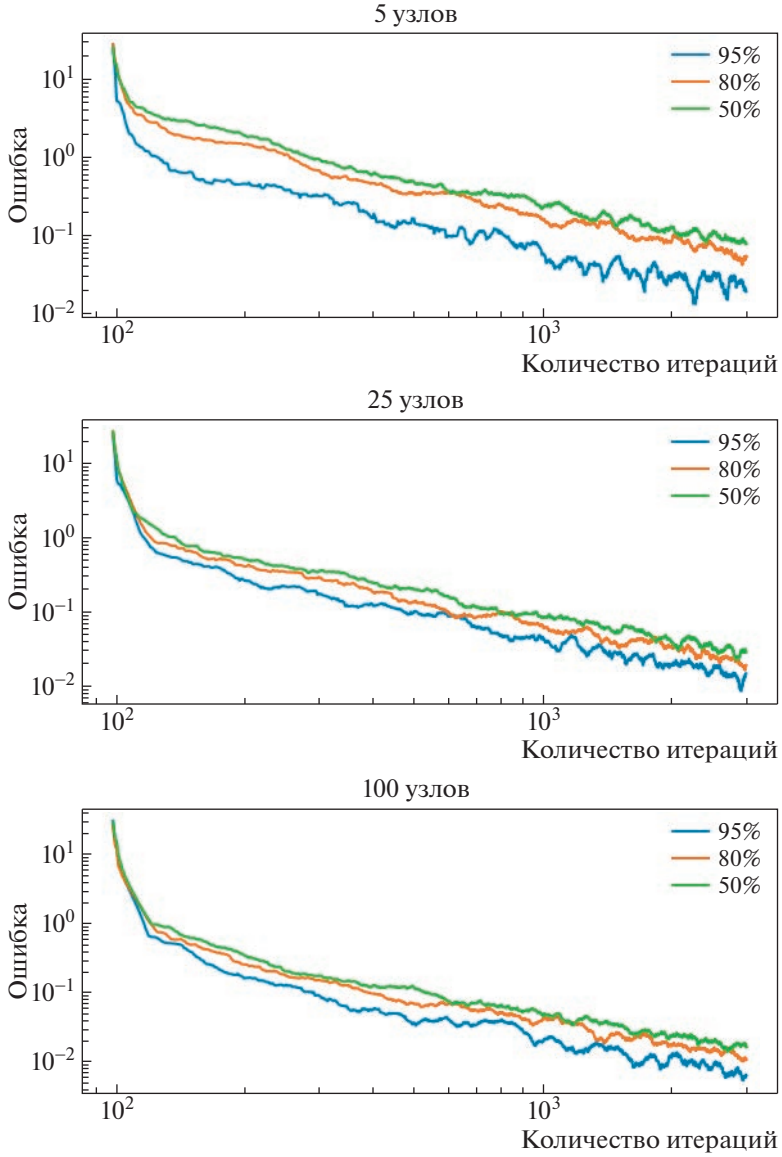

Рис. 1. Синей кривой обозначен график для $p = 0.95$, оранжевым – для $p = 0.8$, и зеленым – для $p = 0.5$. Заметим, что для всех значений вероятностей выполняется $p > \log N/N$, что почти наверное гарантирует связность графа [10].

$$N_{comm} = O\left(\frac{1}{1-\lambda}\frac{L\rho^{-2}}{\varepsilon}\right)$$

коммуникационных шагов и

$$N_{orcl} = O\left(\frac{L\rho^{-2}}{\varepsilon}\right)$$

локальных вызовов линейного минимизационного оракула на каждом узле.

4. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Задача лассо-регрессии (Least Absolute Shrinkage and Selection Operator) – разновидность задачи линейной регрессии, метод регуляризации линейной модели (L1-регуляризация). В LASSO к целевой функции добавляется штраф на сумму абсолютных значений параметров модели, что приводим к тому, что коэффициенты признаком с наименьшей информативностью уменьшаются, или, часто, приравниваются к нулю, что отличает метод L1-регуляризации от классической L2-регуляризации.

Использование регуляризации позволяет произвести отбор наиболее важных признаков модели, сделав ее более интерпретируемой.

Ставится задача так: есть выборка из n наблюдений переменной, значения наблюдений записаны в векторе y, в матрице A – значения призна-

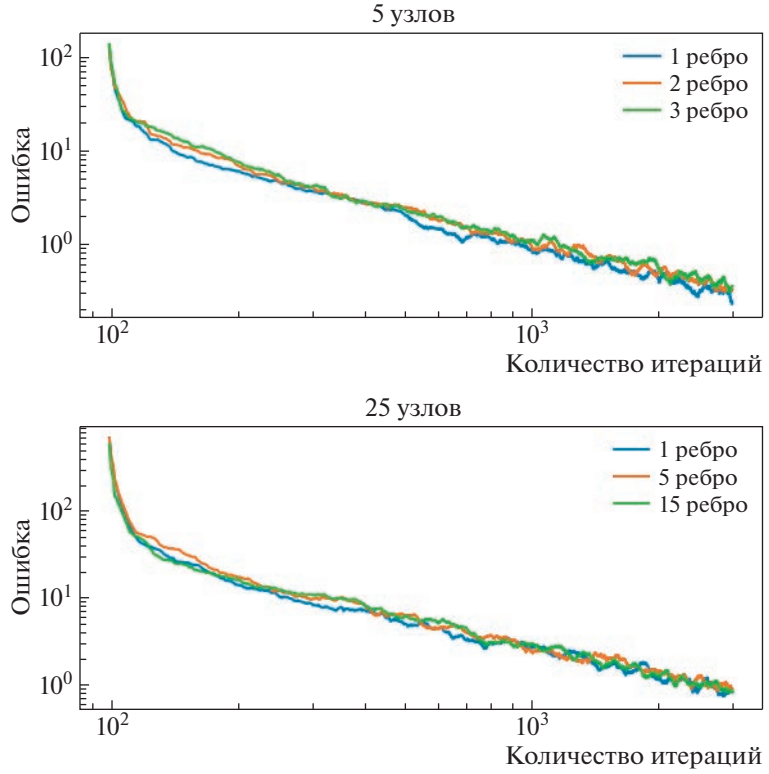

Рис. 2. Изначальный граф сгенерирован с вероятностью появления ребер $p = 0.5$.

ков, в векторе x – параметры (веса) модели, пусть θ – штраф на сложность модели.

Тогда задача выглядит следующим образом:

$$F_{LASSO}(x) = \frac{1}{2}\|y - Ax\|_2^2 + \theta\|x\|_1 \to \min \qquad (4.1)$$

Такая постановка задачи эквивалентна задаче минимизации квадратичного функционала на симплексе, что упрощает ее решение.

$$F_{LASSO}(x) = \frac{1}{2}\|y - Ax\|_2^2 \to \min$$

$$\text{s.t.}\|x\|_1 \leqslant t.$$

Таким образом, действие оракула алгоритма Франк–Вульфа для задачи LASSO можно описать таким образом: алгоритм считает градиент в точке и двигается в сторону, противоположную направлению наибольшей по модулю компоненты градиента.

Опишем гипотезы, которые были проверены при моделировании алгоритма. Во-первых, проверялась зависимость скорости сходимости алгоритма при степенях разреженности графа. Чтобы это сделать, мы воспользовались моделью Эрдеша–Реньи генерации случайных графов, которая заключается в том, что каждое возможное ребро графа генерируется с вероятностью p.

Итак, для разных значений p (а значит и для разной разреженности графов) была измерена скорость сходимости алгоритма для одной и той же задачи для $N = 5$, $N = 25$, $N = 100$ (рис. 1, 2).

Кроме того, была смоделирована ситуация, когда граф, который инициализируется изначально, не генерируется заново на каждой итерации, а меняется слабо, причем количество ребер в нем остается неизменным.

Как видно из графиков, не удается обнаружить зависимости скорости сходимости от количества замененных ребер.

5. ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен метод Франк–Вульфа на переменных во времени графах. С теоретической точки зрения, было рассмотрено два режима изменения графа: детерминированная и стохастическая последовательность графов. Для обоих случаев показано, что алгоритм сходится со скоростью порядка $O(1/t)$, где t – номер итерации. Также были проведены численные эксперименты, подтверждающие теоретические результаты.

ИСТОЧНИК ФИНАНСИРОВАНИЯ

Данная работа поддержана грантом Российского научного фонда (проект № 23-11-00229), https://rscf.ru/en/project/23-11-00229/.

СПИСОК ЛИТЕРАТУРЫ

1. *Braun G., Carderera A., Combettes C.W. Hassani H., Karbasi A. Mokhtari A., Pokutta S.* arXiv (2022) https://arxiv.org/pdf/2211.14103.pdf
2. *Левитин Е.С., Поляк Б.Т.* Методы минимизации при наличии ограничений. Журнал вычислительной математики и математической физики 6.5. 1966. P. 787–823.
3. *Nedic Angelia*. Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. IEEE Signal Processing Magazine 37.3. 2020. P. 92–101.
4. *Forero Pedro A., Alfonso Cano, and Georgios B.* Giannakis. Consensus-based distributed linear support vector machines. Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. 2010.
5. *Gan Lingwen*, *Ufuk Topcu, and Steven H. Low*. Optimal decentralized protocol for electric vehicle charging. IEEE Transactions on Power Systems 28.2. 2012. P. 940–951.
6. *Ram Sundhar Srinivasan*, *Venugopal V. Veeravalli, and Angelia Nedic*. Distributed non-autonomous power control through distributed convex optimization. IEEE INFOCOM 2009. IEEE, 2009.
7. *Ren Wei, and Randal W. Beard.* Distributed consensus in multi-vehicle cooperative control. V. 27. № 2. London: Springer London, 2008.
8. *Rogozin A., Gasnikov A., Beznosikov A., Kovalev D.* Decentralized convex optimization over time-varying graphs: a survey. arXiv (2022) https://arxiv.org/pdf/2210.09719.pdf
9. *Wai Hoi-To et al.* Decentralized FrankвЂ“Wolfe algorithm for convex and nonconvex problems. IEEE Transactions on Automatic Control 62.11. 2017. P. 5522–5537.
10. *Райгородский А.М.* Модели случайных графов и их применения. Труды Московского физико-технического института, 2010.

DECENTRALIZED CONDITIONAL GRADIENT METHOD ON TIME-VARIABLE GRAPHS

R. A. Vedernikov[a], A. V. Rogozin[a], and A. V. Gasnikov[b,c]

[a] *Moscow Institute of Physics and Technology*
Institutskiy per., 9, Moscow region, Dolgoprudny, 141701 Russia
[b] *Institute for Information Transmission Problems of the RAS (Kharkevich Institute)*
Bolshoi Karetny lane, 19, build. 1, Moscow, 127051 Russia
[c] *Caucasian Mathematical Center of the Adyghe State University*
st. Pervomaiskaya, 208, Maykop, Republic of Adygea, 385016 Russia

In this paper, we consider a generalization of the decentralized Frank-Wulff algorithm for network time variables, study the convergence properties of the algorithm, and carry out the corresponding numerical experiments. The changing network is modeled as a deterministic or stochastic sequence of graphs.