Московский физико-технический институт
(национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Кафедра интеллектуальных систем

Богданов Александр Иванович

Применение стохастической аппроксимации нулевого порядка с техникой запоминания в алгоритме Франка-Вульфа

03.03.01 — Прикладные математика и физика

Выпускная квалификационная работа бакалавра

Научный руководитель:
к.ф.-м.н. А. Н. Безносиков

Москва — 2024

Аннотация

В данной работе рассматривается проблема оптимизации "черного ящика". В такой постановке задачи не имеется доступа к градиенту целевой функции, поэтому его необходимо как-то оценить. Предлагается новый способ аппроксимации JAGUAR, который запоминает информацию из предыдущих итераций и требует $\mathcal{O}(1)$ обращений к оракулу. Я реализую эту аппроксимацию для алгоритма Франка-Вольфа и докажу сходимость для выпуклой постановки задачи. Также в данной работе рассматривается стохастическая задача минимизации на множестве Q с шумом в оракуле нулевого порядка, такая постановка довольно непопулярна в литературе, но мы доказали, что JAGUAR-аппроксимация является робастной не только в детерминированных задачах минимизации, но и в стохастическом случае. Я провел эксперименты по сравнению моего градиентного оценщика с уже известными в литературе и подтверждаю доминирование своего метода.

Содержание

1 Введение

Методы без проекций, такие как условный градиент, известный как алгоритм Франка-Вульфа (ФВ)[1], широко используются для решения различных задач оптимизации. В последнее десятилетие методы условного градиента вызывают все больший интерес в сообществе машинного обучения, поскольку во многих случаях вычислительно дешевле решить линейную задачу минимизации на подходящем выпуклом множестве (например, на l_p-шарах или симплексе Δ_d), а затем сделать проекцию на него. [2, 3, 4, 5, 6, 7, 8].

В оригинальной работе Франка-Вульфа [1] авторы использовали истинный градиент в своем алгоритме, однако современные задачи машинного обучения и искусственного интеллекта требуют использования различных оценок градиента, что связано со значительным увеличением размера датасетов и сложности современных моделей. Примерами таких градиентных оценок в алгоритмах типа ФВ являются координатные методы [9, 10, 11] и стохастическая градиентная аппроксимация с батчами [12, 13, 14].

Но иногда встречаются еще более сложные ситуации, когда мы не можем вычислить градиент в общем случае, потому что он недоступен по разным причинам, например, целевая функция не дифференцируема или вычисление градиента вычислительно сложно [15, 16, 17, 18, 19]. Такая постановка называется оптимизацией "черного ящика" [20], и в этом случае мы вынуждены использовать методы оценки градиента нулевого порядка через конечные разности функции цели (иногда с дополнительным шумом) для аппроксимации градиента [21, 22].

За последние годы исследований по теме оптимизации "черного ящика" можно выделить два основных метода аппроксимации градиента с помощью конечных разностей. Первый оценивает градиент в m координатах [23, 24, 25]:

$$\frac{d}{m}\sum_{i\in I}\frac{f(x+\tau e_i)-f(x-\tau e_i)}{2\tau}e_i, \tag{1}$$

где $I \subset \overline{1,d} : |I| = m$, e_i – вектор из стандартного базиса в $\mathbb{R}^d$ и τ – параметр

сглаживания.

Эта конечная разность аппроксимирует градиент в координатах m и требует $\mathcal{O}(m)$ вызовов оракула. Если m мало, то такая оценка будет неточной, если m велико, то на каждой итерации нужно делать много обращений к оракулу нулевого порядка. В случае $m = d$ мы называем этот метод *полная аппроксимация*.

Второй использует в конечной разности не стандартный базис, а случайные вектора e [17, 22, 26, 27]:

$$d\frac{f(x+\tau e) - f(x-\tau e)}{2\tau}e, \tag{2}$$

где e может быть равномерно распределено на l_p-сфере $RS_p^d(1)$, тогда эта схема называется l_p-*сглаживание*. В последних работах авторы обычно используют $p = 1$ [28, 29] или $p = 2$ [30, 31, 32]. Кроме того, e может быть взято из нормального распределения с нулевым средним и единичной ковариационной матрицей [17].

Аппроксимации (1) и (2) имеют очень большую дисперсию или требуют много обращений к нулевому оракулу, поэтому возникает необходимость как-то уменьшить ошибку аппроксимации, не увеличивая при этом количество обращений к нулевому оракулу. В стохастической оптимизации довольно широко используется метод запоминания информации с предыдущих итераций, например, в SVRG [33], SAGA [34], SARAH [35] и SEGA [36] авторы предлагают запоминать градиент с предыдущих итераций для лучшей сходимости метода. Я решил использовать эту технику в задаче оптимизации "черного ящика" и запоминать градиентные аппроксимации из предыдущих итераций для уменьшения размера батча без существенной потери точности.

В этой работе я попытаюсь ответить на следующие вопросы:

- *Можно ли создать метод нулевого порядка, который будет использовать информацию из предыдущих итераций и аппроксимировать истинный градиент так же точно, как и полная аппроксимация* (1), *но потребует* $\mathcal{O}(1)$ *вызовов оракула нулевого порядка?*

- *Можно ли реализовать этот метод аппроксимации в алгоритме Франка-Вольфа для детерминированных и стохастических постановок задач минимизации?*

- *Является ли оценка сходимости этого метода лучше, чем для разностных схем* (1) *и* (2)*?*

В более реалистичной постановке оракул нулевого порядка возвращает зашумленное значение целевой функции, то есть выдает не $f(x)$, а $f(x) + \delta(x)$. В литературе рассматриваются различные виды шума $\delta(\cdot)$: он может быть стохастическим [26, 32, 37, 38] или детерминированным [39, 40, 41, 42, 43, 44]. Поэтому возникает еще один исследовательский вопрос:

- *Как различные типы шума влияют на теоретические гарантии и практические результаты для предложенных мной подходов?*

1.1 Мой вклад

В соответствии с вопросами исследования, мой вклад может быть обобщен следующим образом:

- Я представляю метод `JAGUAR`, который аппроксимирует истинный градиент целевой функции $\nabla f(x)$ в точке x. Использование памяти предыдущих итераций позволяет достичь точности, близкой к полной аппроксимации (1), но `JAGUAR` требует не $\mathcal{O}(d)$, а $\mathcal{O}(1)$ обращений к оракулу нулевого порядка. Сглаживание l_p (2) также требует $\mathcal{O}(1)$ обращения к оракулу, но поскольку в нем нет техники памяти, этот метод имеет большую дисперсию и не является робастным.

- Я доказал теоретические оценки для этого метода (см. раздел 3.1). Мы рассматриваем как детерминированные, так и стохастические шумы в оракуле нулевого порядка. Если первая настройка так или иначе получена в литературе [44, 45], то вторая редко рассматривается авторами, поэтому наш метод подходит для различных задач оптимизации "черного ящика".

- Я внедрил аппроксимацию JAGUAR в алгоритм Франка-Вольфа для стохастических и детерминированных задач минимизации и доказал сходимость в обоих случаях (см. разделы 3.2 и 3.3).

- Я провел несколько вычислительных экспериментов, сравнивая JAGUAR-аппроксимацию с l_2-сглаживанием (2) и полной аппроксимаций (1) на различных задачах минимизации (см. раздел 4).

1.2 Сопутствующие работы

В этом разделе мы сравниваем постановки задач и методы аппроксимации в литературе о методах нулевого порядка в алгоритмах, основанных на Фрэнке-Вулфе. Некоторые авторы считают координатные методы [9] это тоже градиентная аппроксимация, но эти методы используют истинный градиент наблюдаемой функции f, поэтому мы не можем напрямую применить их в оптимизации черного ящика.

Метод l_p-сглаживания не требует дифференцируемости целевой функции, поскольку рассматривает сглаженную версию функции f вида $f_\gamma(x) = \mathbb{E}_e\left[f(x + \gamma e)\right]$. В общем случае метод l_p-сглаживания может аппроксимировать градиент с помощью $\mathcal{O}(1)$ вызовов оракула [43], но он может быть не робастным в постановке Франка-Вульфа, поскольку в [45] авторам приходится собирать большую батч направлений e для достижения сходимости. Отметим, что в [45] рассматривается нестохастический шум.

Полная аппроксимация также используется в литературе [46, 47, 48], но на каждой итерации нам необходимо делать $\mathcal{O}(d)$ вызовов оракула, а поскольку в современных приложениях d огромно, это может быть проблемой. Также этот метод требует гладкости объективной функции f.

В таблице 1 приведено сравнение постановок задач, методов аппроксимации и результатов для них.

Таблица 1: Сопоставление различных методов нулевого порядка и координатных методов ФВ.

Метод	Постановка		Шум		Размер батча	Аппроксимация
	Гладкая	Нулевой порядок	Стахастический	Детерминированный		
ZO-SCGS [45]	✗	✓	✗	✓	$\mathcal{O}\left(1/\varepsilon^2\right)$	l_2-сглаживание (2)
FZFW [47]	✓	✓	✗	✗	$\mathcal{O}\left(\sqrt{d}\right)$	полная аппроксимация (1)
DZOFW [46]	✓	✓	✗	✗	$\mathcal{O}\left(d\right)$	полная аппроксимация (1)
MOST-FW [48]	✓	✓	✗	✗	$\mathcal{O}\left(d\right)$	полная аппроксимация (1)
BCFW [9]	✓	✗	✗	✗	$\mathcal{O}\left(1\right)$	координатный
SSFW [49]	✓	✗	✗	✗	$\mathcal{O}\left(1\right)$	координатный
ФВ с JAGUAR (эта работа)	✓	✓	✓	✓	$\mathcal{O}\left(1\right)$	JAGUAR (Алгоритмы 1 и 4)

2 Постановка задачи

2.1 Франк-Вульф с JAGUAR. Детерминированный случай

В данном разделе рассматривается оптимизационная задача:

$$f^* := \min_{x \in Q} \quad f(x), \tag{3}$$

Предполагается, что доступ есть только к оракулу нулевого порядка, и он возвращает зашумленное значение функции f:

$$f_\delta(x) := f(x) + \delta(x).$$

Несколько предположений, необходимых для анализа.

1. Множество Q – компактное и выпуклое, т.е.

$$\exists\ D > 0 : \forall x, y \in Q \hookrightarrow \|x - y\| \leq D \tag{4}$$

$$\forall\ 0 \leq \alpha \leq 1, \forall x, y \in Q \hookrightarrow \alpha x + (1 - \alpha) y \in Q \tag{5}$$

2. Функция $f(x)$ L-гладкая на множестве Q, т.е.

$$\exists\ L > 0 : \forall x, y \in Q \hookrightarrow \|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|. \tag{6}$$

3. Функция $f(x)$ выпуклая на множестве Q, т.е.

$$\forall x, y \in Q \hookrightarrow f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle. \tag{7}$$

4. Шум $\delta(x)$ оракула ограничен, т.е.

$$\exists\ \Delta > 0 : \forall x \in Q \hookrightarrow |\delta(x)|^2 \leq \Delta^2. \tag{8}$$

2.2 Франк-Вульф с JAGUAR. Стохастический случай

3 Основные результаты

3.1 JAGUAR. Детерминированный случай

We reviewed above the gradient approximation techniques using finite differences (1) and (2). In this section, we introduce new gradient estimation technique JAGUAR, based on the already investigated methods and using memory from previous iterations. We define such additional notation that is used in our algorithm of gradient approximation:

$$\widetilde{\nabla}_i f_\delta(x) := \frac{f_\delta(x + \tau e_i) - f_\delta(x - \tau e_i)}{2\tau} e_i, \tag{9}$$

where e_i is a vector from is standard basis in $\mathbb{R}^d$ as was mentioned above. Now we can present an algorithm of gradient approximation in the point x (Algorithm 1):

Алгоритм 1 JAGUAR gradient approximation. Deterministic case

1: **Input:** $x, h \in \mathbb{R}^d$
2: Sample $i \in \overline{1, d}$ independently and uniform
3: Compute $\widetilde{\nabla}_i f_\delta(x) = \frac{f_\delta(x+\tau e_i) - f_\delta(x-\tau e_i)}{2\tau} e_i$
4: $h = h - h e_i e_i + \widetilde{\nabla}_i f_\delta(x)$

The idea behind the JAGUAR method is similar to well-known variance reduction techniques such as SAGA [34] or SVRG [33]. However, in the zero-order optimization we need to approximate the gradient, therefore, we need to apply the technique of variance reduction to coordinates [36]. Consequently, the JAGUAR method uses the memory of some coordinates of the previous gradients rather than memorizes gradients by batches in past points. There are already works in the literature that combines zero-order optimization and variance reduction, but the essence of these papers is that they change the gradient calculation to the gradient-free approximation (1) in the batch variance reduced algorithms such

as SVRG or SPIDER [50], rather than using variance reduction technique for coordinates as in Algorithm 1.

JAGUAR approximation algorithm can be used with any iterative schemes that, at each step k, return a new point x^k. Using these points, we obtain the sequence h^k in line 4, which serves in a sense as a memory of the gradient components from the past moments. Therefore, it make sense to use h^k as the estimator of the true gradient $\nabla f(x^k)$ in incremental optimization methods. Using the following unified scheme, we can describe such iterative algorithm, that solves the problem (3) (Algorithm 2).

In Algorithm 2, Proc(x^k, grad_est) is some sequence of actions that translates x^k into x^{k+1} by using grad_estimator as true gradient. Now we start to analyze JAGUAR gradient approximation (Algorithm 1). Our goal is to estimate the closeness of the true gradient $\nabla f(x^k)$ and the output of the JAGUAR algorithm h^k at step k.

Алгоритм 2 Iterative algorithm using gradient estimator via JAGUAR

1: **Input:** same as for Proc and h^0
2: **for** $k = 0, 1, 2, ..., N$ **do**
3: $h^{k+1} =$ JAGUAR(x^k, h^k)
4: Proc(x^k, grad_est $= h^{k+1}$)
5: **end for**

Лемма 1. *For x^k and h^k, generated by Algorithm 2, the following inequality holds*

$$H_{k+1} \leq \left(1 - \frac{1}{2d}\right) H_k + 2d\nabla f(x^{k+1}) - \nabla f(x^k)^2 + \nabla f_\delta(x^k) - \nabla f(x^k)^2, \quad (10)$$

where we use notations $H_k := h^k - \nabla f(x^k)^2$ and

$$\widetilde{\nabla} f_\delta(x) := \sum_{i=1}^{d} \frac{f_\delta(x + \tau e_i) - f_\delta(x - \tau e_i)}{2\tau} e_i. \quad (11)$$

For a detailed proof of Lemma 1, see proof of Lemma **??** in Appendix **??** in the case of $\sigma_\nabla^2 = \sigma_f^2 = 0$ (see details in Section 3.3). We do not need any assumptions to satisfy Lemma 1, since in its proof we used only the form of Algorithm 1. That is, the performance of the JAGUAR approximation depends only on the quality of the full approximation $\widetilde{\nabla} f_\delta(x)$ method and the closeness of the points x^{k+1} and x^k, generated by the Algorithm 2. According to Lemma **??** in Appendix **??**, we

can estimate quality of the $\widetilde{\nabla} f_\delta(x)$: under Assumptions 6 and 8 for all $x \in Q$ it holds that

$$\widetilde{\nabla} f_\delta(x) - \nabla f(x)^2 \leq dL^2\tau^2 + \frac{2d\Delta^2}{\tau^2}. \tag{12}$$

Let us analyse the formula (10), and show that using JAGUAR gradient approximation (Algorithm 1) gives us the same estimates as using $\widetilde{\nabla} f_\delta(x)$. Many algorithms of optimization use small enough step size γ, i.e. $\texttt{Proc}(x^k, \texttt{grad\_est}) \approx x^k$. Therefore, we can assume that $\nabla f(x^{k+1}) - \nabla f(x^k)^2 \approx 0$ (for specific choice of γ see Theorem 1 in Section 3.2), then we can unroll (10): $h^k - \nabla f(x^k)^2 \leq h^0 - \nabla f(x^0)^2 e^{-k/2d} + 2\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)^2$. If we consider $k \gg d$, then we can obtain that $h^k - \nabla f(x^k)^2 = \mathcal{O}\left(\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)^2\right)$, i.e. it is the same estimate (12) as for the full approximation estimator (11), however, we now make $\mathcal{O}(1)$ oracle calls at each iteration.

In the next sections, we implement the JAGUAR approximation into the Frank-Wolfe algorithm (see Sections 3.2, 3.3) and in Gradient Descent (see Section **??**).

3.2 Франк-Вульф с JAGUAR. Детерминированный случай

In this section, we introduce the Frank-Wolfe algorithm, that solves the problem (3) using the JAGUAR approximation of the gradient (Algorithm 1) (Algorithm 3).

Using a given form of the Proc function in Algorithm 3, we can unroll results of Lemma 1 to carefully choose step size γ_k.

Алгоритм 3 FW via JAGUAR. Deterministic case

1: **Input:** $x^0 \in Q$, $h^0 = \widetilde{\nabla} f_\delta(x^0)$, γ_k, τ
2: **for** $k = 0, 1, 2, ..., N$ **do**
3: $h^{k+1} = \texttt{JAGUAR}\left(x^k, h^k\right)$
4: $s^k = \arg\min_{x \in Q} \left\langle s, h^{k+1} \right\rangle$
5: $x^{k+1} = x^k + \gamma_k(s^k - x^k)$
6: **end for**

Теорема 1 (Step tuning for FW via JAGUAR (Algorithm 3). Deterministic case)**.** *Consider Assumptions 4 and 6. For h^k, generated by Algorithm 3, we can take $\gamma_k = 4/(k + 8d)$, then the following inequality holds:*

$$h^k - \nabla f(x^k)^2 = \mathcal{O}\left(\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)^2 + \frac{d^2 L^2 D^2}{(k+8d)^2}\right).$$

From Theorem 1 we can conclude that after $\mathcal{O}\left(\frac{\sqrt{d}D}{\tau}\right)$ steps we get the same estimate as in the full-approximation (11). We now explore the convergence of Algorithm 3.

Теорема 2 (Convergence rate of `FW via JAGUAR` (Algorithm 3))**.** *Consider Assumptions 4, 6, 7 and 8. If we take $\gamma_k = 4/(k+8d)$, then `FW via JAGUAR` (Algorithm 3) has the following convergence rate*

$$f(x^k) - f^* = \mathcal{O}\left(\frac{d\max\{LD^2, F_0\}}{N+8d} + \sqrt{d}LD\tau + \frac{\sqrt{d}\Delta D}{\tau}\right),$$

The results of Theorem 2 are matched with results [1, 51] in which the authors used a true gradient and they got the result of the form $f(x^N) - f^* = \mathcal{O}\left(\max\{LD^2; f(x^0) - f^*\}/N\right)$. In the zero-order case, terms of the form $\mathcal{O}\left(\text{poly}(\tau) + \text{poly}\right.$ appear inevitably, since they are crucial for the approximation of the true gradient and always affect the convergence of zero-order methods [39, 41, 52, 53]. The factor d, that appears in our theoretical estimators compared to the first-order result, is related to the zero-order structure of the method.

The results of Theorem 2 can be rewritten as an upper complexity bound on a number of iterates of Algorithm 3, using proper smoothing parameter τ and noise boundary Δ.

Следствие. *Under the conditions of Theorem 2, choosing γ_k, τ, Δ as*

$$\gamma_k = \frac{4}{k+8d},\ \tau = \mathcal{O}\left(\frac{\varepsilon}{\sqrt{d}LD}\right),\ \Delta = \mathcal{O}\left(\frac{\varepsilon^2}{dLD^2}\right),$$

in order to achieve an ε-approximate solution (in terms of $f(x^k) - f^* \leq \varepsilon$) it takes

$$\mathcal{O}\left(\frac{d\max\{LD^2, F_0\}}{\varepsilon}\right) \text{ iterations of Algorithm 3.}$$

In Corollary 3.2, the dependence $\Delta(\varepsilon)$ was obtained, which may seem incorrect, because usually the maximum noise is given to us by the nature and we cannot reduce it. In this case, we should rewrite the dependence in the form $\varepsilon = \varepsilon(\Delta)$ and accordingly τ and N start to depend on Δ, not on ε.

Следствие. *Under the conditions of Theorem 2, choosing* $\gamma_k, \tau, \varepsilon$ *as*

$$\gamma_k = \frac{4}{k+8d}, \ \tau = \mathcal{O}\left(\sqrt{\Delta/L}\right), \ \varepsilon = \mathcal{O}\left(\sqrt{dLD^2\Delta}\right),$$

in order to achieve an ε*-approximate solution (in terms of* $f(x^k) - f^* \leq \varepsilon$*) it takes*

$$\mathcal{O}\left(\frac{\sqrt{d}\max\{LD^2, F_0\}}{\sqrt{LD^2\Delta}}\right) \text{ iterations of Algorithm 3.}$$

In the rest of the corollaries in this paper, we will write the dependence $\Delta(\varepsilon)$ for the convenience of presentation, but they can always be rewritten in terms of $\varepsilon(\Delta)$.

For a detailed proof of Theorems 1, 2 and Corollaries 3.2, 3.2, see Appendix **??**.

3.3 Франк-Вульф с JAGUAR. Стохастический случай

In this section, we consider the stochastic version of the problem (3):

$$f(x) := \mathbb{E}_{\xi\sim\pi}\left[f(x,\xi)\right], \tag{13}$$

where ξ is random vector from usually unknown distribution π. For this problem, we can not use the values of the function $f(x)$ in the difference schemes, since only $f(x,\xi)$ is available. We again assume that we do not have access to the true value of the gradient $\nabla f(x,\xi)$, and zero-order oracle returns the noisy value of the function $f(x,\xi)$: $f_\delta(x,\xi) := f(x,\xi) + \delta(x,\xi)$.

In the stochastic setup (13), two versions of the differences of scheme (9) appear. First one is called two point feedback (TPF) [26, 31, 32, 54, 55]. In this case, we define such gradient approximations of the function $f(x)$:

$$\widetilde{\nabla}_i f_\delta(x,\xi) := \frac{f_\delta(x+\tau e_i,\xi) - f_\delta(x-\tau e_i,\xi)}{2\tau} e_i. \tag{14}$$

Second one is called one point feedback (OPF) [30, 38, 56, 57, 58]. In this case, we define slightly different gradient approximation of the function $f(x)$:

$$\widetilde{\nabla}_i f_\delta(x,\xi^\pm) := \frac{f_\delta(x+\tau e_i,\xi^+) - f_\delta(x-\tau e_i,\xi^-)}{2\tau} e_i. \tag{15}$$

The key difference between approximations (14) and (15) is that scheme (14) is more accurate, but it is difficult to implement in practice, because we have to get the same realization of ξ at two different points $x+\tau e$ and $x-\tau e$, therefore, the scheme (15) is more interesting from a practical point of view. To simplify further, we consider that in the two point feedback case (14) we have the same inscription as for the one point feedback (15), but only $\xi^+ = \xi^- = \xi$. We provide several assumptions required for the analysis.

[Smoothness] The functions $f(x,\xi)$ are $L(\xi)$-smooth on the set Q.

We also assume that exists constant L such that $L^2 := L(\xi)^2$.

If Assumption 3.3 holds, then function $f(x)$ is L-smooth on the set Q, since for all $x, y \in Q$ holds that $\nabla f(x) - \nabla f(y)^2 = \nabla f(x,\xi) - \nabla f(y,\xi)^2 \leq \nabla f(x,\xi) - \nabla f(y,\xi)^2 \leq L^2 x - y^2$.

Because the zero-order oracle returns to us noisy values of the function $f(x,\xi)$ we make common assumption on this noise.

[Bounded oracle noise] The noise in the oracle is bounded by some constant $\Delta > 0$, i.e. $\exists\ \Delta > 0 :\ \forall x \in Q \hookrightarrow |\delta(x,\xi)|^2 \leq \Delta^2$.

If Assumption 3.3 holds, then if we define $\delta(x) := \delta(x,\xi)$, then it holds that $|\delta(x)|^2 \leq \Delta^2$, since $|\delta(x)|^2 = |\delta(x,\xi)|^2 \leq |\delta(x,\xi)|^2 \leq \Delta^2$.

Now we present two assumptions that are needed only in the stochastic case.

[Bounded second moment of gradient] The second moment of the $\nabla f(x,\xi)$ is bounded, i.e.

$$\exists\ \sigma_\nabla \geq 0 :\ \forall x \in Q \hookrightarrow \nabla f(x,\xi) - \nabla f(x)^2 \leq \sigma_\nabla^2.$$

[Bounded second moment of function] The second moment of the $f(x,\xi)$ is

bounded, i.e.

$$\exists\ \sigma_f \geq 0:\ \forall x \in Q \hookrightarrow |f(x,\xi) - f(x)|^2 \leq \sigma_f^2.$$

In the two point feedback case (14), we do not need Assumption 3.3, therefore, for simplicity of future exposition we assume that in this case Assumption 3.3 is fulfilled with $\sigma_f = 0$.

Now we can present the Frank-Wolfe algorithm, that solves the problem (3) + (13) using `JAGUAR` gradient approximation.

Алгоритм 4 `FW via JAGUAR`. Stochastic case

1: **Input:** $x^0 \in Q$, $h^0 = g^0 = \widetilde{\nabla} f_\delta(x^0, \xi^{\pm}_{\overline{1,d}})$, γ_k, η_k, τ

2: **for** $k = 0, 1, 2, ..., N$ **do**

3: Sample $i_k \in \overline{1,d}$ independently and uniform

4: Sample 2 realizations of ξ: ξ_k^+ and ξ_k^- independently (in TPF $\xi_k^+ = \xi_k^-$)

5: Compute $\widetilde{\nabla}_{i_k} f_\delta(x^k, \xi_k^{\pm}) = \frac{f_\delta(x^k + \tau e_{i_k}, \xi_k^+) - f_\delta(x^k - \tau e_{i_k}, \xi_k^-)}{2\tau} e_{i_k}$

6: $h^{k+1} = h^k - h^k e_{i_k} e_{i_k} + \widetilde{\nabla}_{i_k} f_\delta(x^k, \xi_k^+, \xi_k^-)$

7: $\rho^k = h^k - d \cdot h^k e_{i_k} e_{i_k} + d \cdot \widetilde{\nabla}_{i_k} f_\delta(x^k, \xi_k^+, \xi_k^-)$

8: $g^{k+1} = (1 - \eta_k) g^k + \eta_k \rho^k$

9: $s^k = \arg\min_{s \in Q} \left\langle s, g^{k+1} \right\rangle$

10: $x^{k+1} = x^k + \gamma_k (s^k - x^k)$

11: **end for**

In Input of Algorithm 4, we use a notation:

$$\widetilde{\nabla} f_\delta(x, \xi^{\pm}_{\overline{1,d}}) := \sum_{i=1}^{d} \frac{f_\delta(x + \tau e_i, \xi_i^+) - f_\delta(x - \tau e_i, \xi_i^-)}{2\tau} e_i.$$

In the two point feedback case (14), $\xi_i^+ = \xi_i^-$.

Algorithm 4 is similar to `FW via JAGUAR` in the deterministic case (Algorithm 3), but in lines 7 and 8 we use SEGA [36] and momentum [59] parts in order to convergence in the stochastic case.

• We need SEGA part [36] ρ_k in Algorithm 4, because in the stochastic case, we care about the "unbiased"property (see proof of Lemma **??** in Appendix **??**), i.e.

$$\mathbb{E}_k[\rho^k] = \widetilde{\nabla} f_\delta(x^k) := \sum_{i=1}^{d} \frac{f_\delta(x+\tau e_i) - f_\delta(x-\tau e_i)}{2\tau} e_i,$$

where $\mathbb{E}_k[\cdot]$ is a conditional mathematical expectation on a step k. Using the SEGA part ρ^k deteriorates our estimates by a factor of d compared to using h^k as a gradient approximation (see Lemmas **??** and **??** in Appendix **??**), but we have to accept this factor.

• We need momentum part [59] η_k in Algorithm 4, because in evaluating the expression of $\widetilde{\nabla} f_\delta(x, \xi^{\pm}_{1,d}) - \nabla f(x)^2$ in the stochastic case appear expressions containing σ_∇^2 and σ_f^2 and they interfere with convergence (see Lemma **??** in Appendix **??**). This is common issue in the stochastic Frank-Wolfe-based methods (see [59]).

We can provide a theorem, similar to Theorem 1, where we carefully choose step sizes γ_k and η_k.

Теорема 3 (Step tuning for `FW via JAGUAR`. Stochastic case)**.** *Consider Assumptions 4, 3.3, 3.3, 3.3 and 3.3 in the one point feedback case. For x^k generated by Algorithm 4, we can take*

$$\gamma_k = \frac{4}{k + 8d^{3/2}} \quad \text{and} \quad \eta_k = \frac{4}{(k+8d^{3/2})^{2/3}},$$

then, the following inequality holds:

$$G_k = \mathcal{O}\left(\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)^2 + \frac{L^2 D^2 + d^2\sigma_f^2/\tau^2 + d^2\sigma_\nabla^2}{(k+8d^{3/2})^{2/3}}\right),$$

where we use the notation $G_k := g^k - \nabla f(x^k)^2$. In the two point feedback case, $\sigma_f^2 = 0$.

We obtain worse estimates compared to the deterministic case in Theorem 1, since we consider a more complicated setup. We now explore the convergence of Algorithm 4.

Теорема 4 (Convergence rate of `FW via JAGUAR` (Algorithm 4). Stochastic case)**.** *Consider Assumptions 4, 7, 3.3, 3.3, 3.3 and 3.3 in the one point feedback case. We can take*

$$\gamma_k = \frac{4}{k + 8d^{3/2}} \quad \text{and} \quad \eta_k = \frac{4}{(k + 8d^{3/2})^{2/3}},$$

then we `FW via JAGUAR` *(Algorithm 4) has the following convergence rate*

$$F_N = \mathcal{O}\left(\frac{LD^2 + d\sigma_f D/\tau + d\sigma_\nabla D + \sqrt{d}F_0}{(N + 8d^{3/2})^{1/3}} + \sqrt{d}LD\tau + \frac{\sqrt{d}\Delta D}{\tau}\right),$$

where we use the notation $F_k := f(x^k) - f^*$*. In the two point feedback case,* $\sigma_f^2 = 0$*.*

Следствие. *Under the conditions of Theorem 4, choosing* $\gamma_k, \eta_k, \tau, \Delta$ *as*

$$\gamma_k = \frac{4}{k + 8d^{3/2}},\ \eta_k = \frac{4}{(k + 8d^{3/2})^{2/3}},\ \tau = \mathcal{O}\left(\frac{\varepsilon}{\sqrt{d}LD}\right),\ \Delta = \mathcal{O}\left(\frac{\varepsilon^2}{dLD^2}\right),$$

in order to achieve an ε*-approximate solution (in terms of* $f(x^N) - f^* \leq \varepsilon$*) it takes*

$$\mathcal{O}\left(\max\left\{\left[\frac{LD^2 + d\sigma_\nabla D + \sqrt{d}(f(x^0) - f^*)}{\varepsilon}\right]^3; \frac{d^{9/2}\sigma_f^3 L^3 D^6}{\varepsilon^6}\right\}\right) \text{ iterations of Algorithm}$$

In the two point feedback case, $\sigma_f^2 = 0$ *and the last equation takes form*

$$\mathcal{O}\left(\left[\frac{LD^2 + d\sigma_\nabla D + \sqrt{d}(f(x^0) - f^*)}{\varepsilon}\right]^3\right).$$

Since we used SEGA and momentum parts in `JAGUAR` approximation algorithm (Algorithm 4) we do not get the same convergence rate as in Theorems 1 and 2 even when we switch from stochastic to deterministic setups, i.e., when setting

$\sigma_\Delta = \sigma_f = 0$ in Theorems 3 and 4. The same problems arise in the first-order case [13, 59], it is due to the difficulties of implementing the stochastic gradient in FW-type algorithms.

We can apply the deterministic `JAGUAR` method (Algorithm 1) to the stochastic problem (13) and obtain the same estimates as in Theorems 1 and 2, only the smoothed term of the form $\mathcal{O}\left(\text{poly}(\tau) + \text{poly}(\Delta/\tau)\right)$ will contain summands of the form $\mathcal{O}\left(\text{poly}(\sigma_\Delta^2) + \text{poly}(\sigma_f^2/\tau)\right)$ Therefore, if $\sigma_\Delta^2, \sigma_f^2 \sim \Delta$, then deterministic Algorithm 1 is suitable for the stochastic problem (13). However, this means that we need to use big batches, therefore, we forced to use SEGA and momentum parts in the `JAGUAR` approximation.

For a detailed proof of Theorem 3, see Appendix **??**, for Theorem 4 and Corollary 3.3, see Appendix **??**.

4 Вычислительный эксперимент

В этом разделе представлены результаты экспериментов по применению аппроксимации JAGUAR в различных задачах оптимизации "черного ящика". Результаты включают оптимизацию с помощью алгоритмов Франка-Вульфа как для детерминированного, так и стохастического случаев. Технические подробности и дополнительные эксперименты представлены в Приложении ??.

4.1 Постановка эксперимента

Я рассмотрел задачу классификации моделью логистической регрессии с регуляризацией L_2 на множестве Q вида:

$$\min_{w \in Q} f(w) = \frac{1}{m} \sum_{k=1}^{m} \log\left(1 + \exp\left(-\mathbf{y}_k \cdot (\mathbf{X}\mathbf{w})_k\right)\right) + \lambda \|\mathbf{w}\|_2^2, \tag{16}$$

где коэффициент регуляризации $\lambda = 0.05$. А также квадратичную задачу:

$$\min_{w \in Q} f(w) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2. \tag{17}$$

В качестве минимизирующего множества Q рассматриваются симплекс Δ_d и L_2-шар.

Для логистической регрессии (16) использовались данные датасета mushrooms из библиотеки LibSVM [60], а для квадратичной задачи (17) – синтетические данные. Было показано, что использование JAGUAR работает лучше, чем использование базовых оценок градиентов: l_2-сглаживания (2) и полной аппроксимации (1).

4.2 Детерминированный Франк-Вульф

В этом разделе предполагаем, что есть шум в виде округления. На рисунке 1 показана сходимость детерминированного алгоритма ФВ с аппроксимацией

JAGUAR (алгоритм 3). Алгоритм значительно превосходит базовые аппроксимации, эти наблюдения подтверждают наши теоретические выводы. В этом разделе можно увидеть результаты только для $Q = \Delta_d$, эксперименты на L_2-шаре можно найти в приложении **??**.

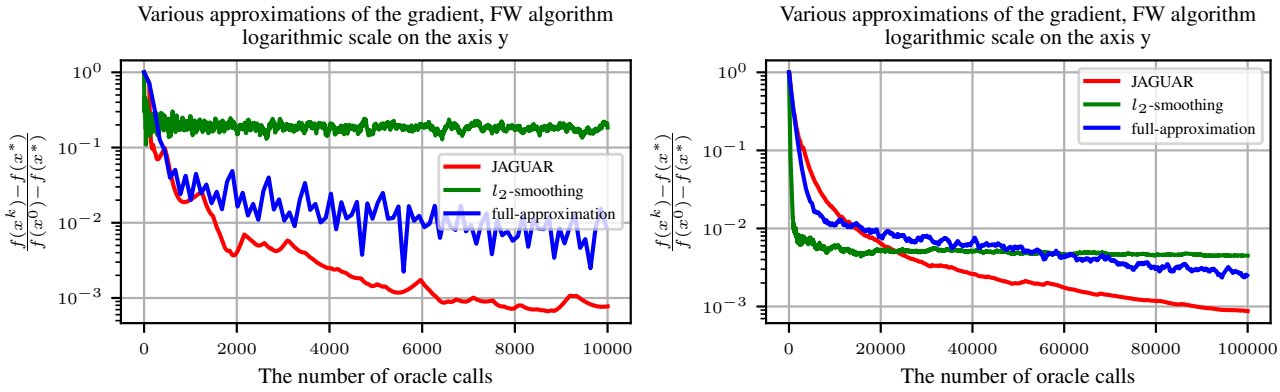

Рис. 1: Сходимость алгоритма ФВ в логарифмической задаче (слева) и квадратичной задаче (справа) на множестве $Q = \Delta_d$.

4.3 Стохастический Франк-Вульф

В этом разделе рассматривается только квадратичная задача минимизации (17), однако предполагается, что есть стохастический шум в виде $\delta(x, \xi) = \xi \sim \mathcal{N}(0, \sigma^2)$, где $\sigma^2 = 0.1$. На рисунке 1 показана сходимость стохастического алгоритма ФВ с аппроксимацией JAGUAR (алгоритм 4). Случай ДОС превосходит базовые алгоритмы с большим отрывом, и эти наблюдения подтверждают наши теоретические выводы. В этом разделе можно увидеть результаты для $Q = \Delta_d$ и L_2-шара.

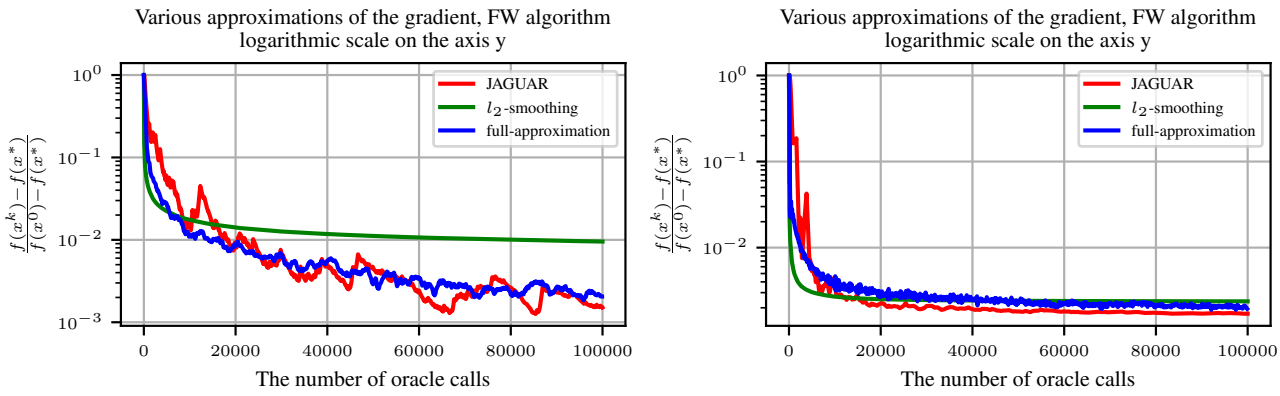

Рис. 2: Сходимость алгоритма ФВ на квадратичной задаче (17) со стохастическим шумом на множествах $Q = \Delta_d$ (слева) и L_2-шаре (справа).

Картинки будут переделаны с подписями на русском и в другом стиле

5 Заключение

Пока что формулируется

Список литературы

[1] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[2] Larry J LeBlanc, Richard V Helgason, and David E Boyce. Improved efficiency of the frank-wolfe algorithm for convex network programs. *Transportation Science*, 19(4):445–462, 1985.

[3] Martin Jaggi. Sparse convex optimization methods for machine learning. 2011.

[4] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[5] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[6] Donald Goldfarb, Garud Iyengar, and Chaoxu Zhou. Linear convergence of stochastic frank wolfe variants. In *Artificial Intelligence and Statistics*, pages 1066–1074. PMLR, 2017.

[7] Ali Dadras, Karthik Prakhya, and Alp Yurtsever. Federated frank-wolfe algorithm. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[8] Robert M Freund, Paul Grigas, and Rahul Mazumder. An extended frank–wolfe method with "in-face" directions, and its application to low-rank matrix completion. *SIAM Journal on optimization*, 27(1):319–346, 2017.

[9] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *International Conference on Machine Learning*, pages 53–61. PMLR, 2013.

[10] Yu-Xiang Wang, Veeranjaneyulu Sadhanala, Wei Dai, Willie Neiswanger, Suvrit Sra, and Eric Xing. Parallel and distributed block-coordinate frank-

wolfe algorithms. In *International Conference on Machine Learning*, pages 1548–1557. PMLR, 2016.

[11] Anton Osokin, Jean-Baptiste Alayrac, Isabella Lukasewitz, Puneet Dokania, and Simon Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. In *international conference on machine learning*, pages 593–602. PMLR, 2016.

[12] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th annual Allerton conference on communication, control, and computing (Allerton)*, pages 1244–1251. IEEE, 2016.

[13] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, pages 4012–4023. PMLR, 2020.

[14] Haihao Lu and Robert M Freund. Generalized stochastic frank–wolfe algorithm with stochastic "substitute" gradient for structured convex optimization. *Mathematical Programming*, 187(1):317–349, 2021.

[15] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd international conference on Machine learning*, pages 896–903, 2005.

[16] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[17] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

[18] Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pages 970–978. PMLR, 2018.

[19] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International conference on machine learning*, pages 1467–1476. PMLR, 2018.

[20] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. *Advances in neural information processing systems*, 28, 2015.

[21] Yu. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. doi:10.1137/100802001. URL https://doi.org/10.1137/100802001.

[22] John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2): 674–701, 2012.

[23] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1):1–38, 2014.

[24] Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.

[25] Yurii Nesterov and Sebastian U. Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017. doi:10.1137/16M1060182. URL https://doi.org/10.1137/16M1060182.

[26] Alexander Gasnikov, Anton Novitskii, Vasilii Novitskii, Farshed Abdukhakimov, Dmitry Kamzolov, Aleksandr Beznosikov, Martin Takac,

Pavel Dvurechensky, and Bin Gu. The power of first-order smooth optimization for black-box non-smooth problems. In *International Conference on Machine Learning*, pages 7241–7265. PMLR, 2022.

[27] Alexander Gasnikov, Darina Dvinskikh, Pavel Dvurechensky, Eduard Gorbunov, Aleksander Beznosikov, and Alexander Lobanov. Randomized gradient-free methods in convex optimization. *arXiv preprint arXiv:2211.13566*, 2022.

[28] Alexander Gasnikov, Anastasia Lagunovskaya, Ilnura Usmanova, and Fedor Fedorenko. Gradient-free proximal methods with inexact oracle for convex stochastic nonsmooth optimization problems on the simplex. *Automation and Remote Control*, 77:2018–2034, 2016.

[29] Arya Akhavan, Evgenii Chzhen, Massimiliano Pontil, and Alexandre Tsybakov. A gradient estimator via l1-randomization for online zero-order optimization with two point feedback. *Advances in Neural Information Processing Systems*, 35:7685–7696, 2022.

[30] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[31] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

[32] Eduard Gorbunov, Pavel Dvurechensky, and Alexander Gasnikov. An accelerated method for derivative-free smooth stochastic convex optimization. *SIAM Journal on Optimization*, 32(2):1210–1238, 2022. doi:10.1137/19M1259225. URL https://doi.org/10.1137/19M1259225.

[33] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

[34] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

[35] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

[36] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. Sega: Variance reduction via gradient sketching. *Advances in Neural Information Processing Systems*, 31, 2018.

[37] Francis Bach and Vianney Perchet. Highly-smooth zero-th order online optimization. In *Conference on Learning Theory*, pages 257–283. PMLR, 2016.

[38] Arya Akhavan, Massimiliano Pontil, and Alexandre Tsybakov. Exploiting higher order smoothness in derivative-free optimization and continuous bandits. *Advances in Neural Information Processing Systems*, 33:9017–9027, 2020.

[39] Andrej Risteski and Yuanzhi Li. Algorithms and matching lower bounds for approximately-convex optimization. *Advances in Neural Information Processing Systems*, 29, 2016.

[40] Lev Bogolubsky, Pavel Dvurechenskii, Alexander Gasnikov, Gleb Gusev, Yurii Nesterov, Andrei M Raigorodskii, Aleksey Tikhonov, and Maksim Zhukovskii. Learning supervised pagerank with gradient-based and gradient-free optimization methods. *Advances in neural information processing systems*, 29, 2016.

[41] Anit Kumar Sahu, Dusan Jakovetic, Dragana Bajovic, and Soummya Kar. Distributed zeroth order optimization over random networks: A kiefer-wolfowitz stochastic approximation approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4951–4958. IEEE, 2018.

[42] Anastasia Sergeevna Bayandina, Alexander V Gasnikov, and Anastasia A Lagunovskaya. Gradient-free two-point methods for solving stochastic nonsmooth convex optimization problems with small non-random noises. *Automation and Remote Control*, 79:1399–1408, 2018.

[43] Darina Dvinskikh, Vladislav Tominin, Iaroslav Tominin, and Alexander Gasnikov. Noisy zeroth-order optimization for non-smooth saddle point problems. In *International Conference on Mathematical Optimization Theory and Operations Research*, pages 18–33. Springer, 2022.

[44] Aleksandr Lobanov, Andrew Veprikov, Georgiy Konin, Aleksandr Beznosikov, Alexander Gasnikov, and Dmitry Kovalev. Non-smooth setting of stochastic decentralized convex optimization problem over time-varying graphs. *Computational Management Science*, 20(1):48, 2023.

[45] Aleksandr Lobanov, Anton Anikin, Alexander Gasnikov, Alexander Gornov, and Sergey Chukanov. Zero-order stochastic conditional gradient sliding method for non-smooth convex optimization. *arXiv preprint arXiv:2303.02778*, 2023.

[46] Anit Kumar Sahu, Manzil Zaheer, and Soummya Kar. Towards gradient free and projection free stochastic optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3468–3477. PMLR, 2019.

[47] Hongchang Gao and Heng Huang. Can stochastic zeroth-order frank-wolfe method converge faster for non-convex problems? In *International conference on machine learning*, pages 3377–3386. PMLR, 2020.

[48] Zeeshan Akhtar and Ketan Rajawat. Zeroth and first order stochastic frank-wolfe algorithms for constrained optimization. *IEEE Transactions on Signal Processing*, 70:2119–2135, 2022.

[49] Aleksandr Beznosikov, David Dobre, and Gauthier Gidel. Sarah frank-wolfe:

Methods for constrained optimization with best rates and practical features. *arXiv preprint arXiv:2304.11737*, 2023.

[50] Kaiyi Ji, Zhe Wang, Yi Zhou, and Yingbin Liang. Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization. In *International conference on machine learning*, pages 3100–3109. PMLR, 2019.

[51] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013.

[52] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[53] Aleksandr Beznosikov, Eduard Gorbunov, and Alexander Gasnikov. Derivative-free method for composite optimization with applications to decentralized distributed optimization. *IFAC-PapersOnLine*, 53(2):4038–4043, 2020.

[54] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[55] Aleksandr Beznosikov, Abdurakhmon Sadiev, and Alexander Gasnikov. Gradient-free methods with inexact oracle for convex-concave stochastic saddle-point problem. In *International Conference on Mathematical Optimization Theory and Operations Research*, pages 105–119. Springer, 2020.

[56] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

[57] Alexander V Gasnikov, Ekaterina A Krymova, Anastasia A Lagunovskaya, Ilnura N Usmanova, and Fedor A Fedorenko. Stochastic online optimization.

single-point and multi-point non-linear multi-armed bandits. convex and strongly-convex case. *Automation and remote control*, 78:224–234, 2017.

[58] Aleksandr Beznosikov, Vasilii Novitskii, and Alexander Gasnikov. One-point gradient-free methods for smooth and non-smooth saddle-point problems. In *Mathematical Optimization Theory and Operations Research: 20th International Conference, MOTOR 2021, Irkutsk, Russia, July 5–10, 2021, Proceedings 20*, pages 144–158. Springer, 2021.

[59] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *The Journal of Machine Learning Research*, 21(1):4232–4280, 2020.

[60] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2 (3):1–27, 2011.

6 Приложение

Оно есть, но пока на английском, поэтому пока не здесь.