ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 1, с. 20–31

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

УДК 519.853.62

УСКОРЕННЫЙ МЕТААЛГОРИТМ ДЛЯ ЗАДАЧ ВЫПУКЛОЙ ОПТИМИЗАЦИИ[1)]

© 2021 г. А. В. Гасников[1,2], Д. М. Двинских[2,1,3], П. Е. Двуреченский[3,2], Д. И. Камзолов[1,*], В. В. Матюхин[1], Д. А. Пасечнюк[1], Н. К. Тупица[1], А. В. Чернов[1]

[1] 141701 Долгопрудный, М.о., Институтский пер., 9, Московский физико-технический институт (национальный исследовательский университет), Россия

[2] 127051 Москва, Большой Каретный пер., 19, стр. 1, Институт проблем передачи информации им. А.А. Харкевича РАН, Россия

[3] Институт прикладного анализа и стохастики им. К. Вейерштрасса, Берлин, Германия

**e-mail: kamzolov.dmitry@phystech.edu*

Поступила в редакцию 18.04.2020 г.
Переработанный вариант 16.06.2020 г.
Принята к публикации 18.09.2020 г.

Предлагается оболочка, названная “ускоренный метаалгоритм”, которая позволяет единообразно получать ускоренные методы решения задач выпуклой безусловной минимизации в различных постановках на базе неускоренных вариантов. В качестве приложений приводятся квазиоптимальные алгоритмы для минимизации гладких функций с липшицевыми производными произвольного порядка, а также для решения гладких минимаксных задач. Предложенная оболочка является более общей, чем существующие, а также позволяет получать лучшие оценки скорости сходимости и практическую эффективность для ряда постановок задач. Библ. 26. Фиг. 2.

Ключевые слова: выпуклая оптимизация, проксимальный ускоренный метод, тензорные методы, неточный оракул, слайдинг, каталист.

DOI: 10.31857/S0044466921010051

1. ВВЕДЕНИЕ

В последние 15 лет в численных методах гладкой выпуклой оптимизации преобладают так называемые ускоренные методы. Прообразом таких методов является метод тяжелого шарика Б.Т. Поляка и моментный метод Ю.Е. Нестерова (см. [1], [2]). Оказалось, что для многих задач гладкой выпуклой оптимизации оптимальные методы (с точки зрения числа вычислений градиента функции; в общем случае, старших производных) могут быть найдены среди ускоренных методов (см. [1]–[3]). Появилось огромное число работ, в которых предлагаются различные варианты ускоренных методов для разных классов задач (см., например, обзор литературы в [1], [3]). Каждый раз процедура ускорения принимала свою причудливую форму. Естественно, возникло желание как-то унифицировать все это. В 2015 г. это было сделано для широкого класса (рандомизированных) градиентных методов с помощью проксимальной ускоренной оболочки, названной Каталист (см. [4]). (Здесь и далее в качестве названий подходов/алгоритмов иногда будут использоваться англицизмы. Дело в том, что дословный перевод исходно английских выражений на русский язык может только запутывать дело. Отметим также, что под “проксимальной оболочкой” здесь и далее имеется в виду просто проксимальный алгоритм. Слово “оболочка” подразумевает, что в проксимальном алгоритме на каждой итерации имеется своя внутренняя (вспомогательная) задача оптимизации, которую, как правило, нельзя решить аналитически. Ее нужно решать численно. Поэтому внешний проксимальный метод можно по-

[1)]Работа А.В. Гасникова выполнена при финансовой поддержке РФФИ (код проекта 18-31-20005 мол_а_вед в п. 2), работа Д.И. Камзолова выаолнена при финансовой поддержке РФФИ (код проекта 19-31-90170). Аспиранты в п. 3, работа П.Е. Двуреченского выаолнена при финансовой поддержке РФФИ (код проекта 18-29-03071 мк в п. 3). Работа Д.М. Двинских и В.В. Матюхина выполнена при финансовой поддержке Минобрнауки РФ (госзадание № 075-00337-20-03, номер проекта 0714-2020-0005).

нимать как “оболочку” для метода, использующегося для решения внутренней задачи.) С 2013 г. данные результаты стали активно переноситься на тензорные методы (использующие старшие производные) (см. [5]–[8]). В самое последнее время предпринимаются попытки унификации процедур ускорения для седловых задач и задач со структурой (композитных задач) (см. [9]–12]). Во всех этих направлениях по-прежнему использовалось значительное разнообразие ускоренных проксимальных оболочек (см. [1], [4]–[8], [10], [11], [13]– [16]). Метод из данной работы будет во многом базироваться на схеме из [14]. (Строго говоря, это даже не метод (алгоритм), а скорее оболочка (в смысле, определенном выше). В данной статье было выбрано название “ускоренный метаалгоритм”. Первое слово поясняет цель разрабатываемой оболочки – ускорение метода, использующегося в качестве базового (решающего внутреннюю задачу). Однако, в отличие от стандартной (ускоренной) оболочки, в предложенной в данной статье оболочке все же в ряде важных случаев вспомогательная задача решается аналитически и, стало быть, говорить об этой оболочке, как “оболочке”, а не как об обычном алгоритме, не совсем корректно. Поэтому было решено использовать более нейтральное в этом смысле слово – “метаалгоритм”.)

В данной работе показывается, что достаточно изучить всего одну ускоренную проксимальную оболочку, которая позволяет получать все известные нам ускоренные методы для задач гладкой выпуклой безусловной оптимизации. Причем в ряде случаев предложенный ускоренный метаалгоритм позволяет убирать логарифмические зазоры в оценках сложности (по сравнению с нижними оценками), имевшие место в предыдущих подходах.

2. ОСНОВНЫЕ РЕЗУЛЬТАТЫ

Рассмотрим следующую задачу (x_* – решение задачи):

$$\min_{x\in\mathbb{R}^d}\{F(x) \coloneqq f(x) + g(x)\}, \tag{1}$$

где f и g – выпуклые функции.

Везде в дальнейшем под $\|\cdot\|$ будем понимать обычную евклидову норму в пространстве $\mathbb{R}^d$,

$$D^k f(x)[h]^k = \sum_{i_1,\ldots,i_d\geq 0:\ \sum_{j=1}^d i_j = k} \frac{\partial^k f(x)}{\partial x_1^{i_1}\ldots\partial x_d^{i_d}} h_1^{i_1}\cdot\ldots\cdot h_d^{i_d},$$

$$\left\|D^k f(x)\right\| = \max_{\|h\|\leq 1}\left\|D^k f(x)[h]^k\right\|.$$

Будем считать, что f имеет липшицевы производные порядка p ($p\in\mathbb{N}$):

$$\left\|D^p f(x) - D^p f(y)\right\| \leq L_{p,f}\|x-y\|. \tag{2}$$

Здесь и далее (см., например, (7)) можно считать, что $x, y\in\mathbb{R}^d$ принадлежат евклидову шару с центром в точке x_* и радиусом $O\left(\left\|x_0 - x_*\right\|\right)$, где x_0 – точка старта (см. [6]).

Введем аппроксимацию рядом Тейлора функции f:

$$\Omega_p(f,x;y) = f(x) + \sum_{k=1}^p \frac{1}{k!} D^k f(x)[y-x]^k, \quad y\in\mathbb{R}^d.$$

Заметим, что из (2) следует (см. [17]), что

$$\left|f(y) - \Omega_p(f,x;y)\right| \leq \frac{L_{p,f}}{(p+1)!}\|y-x\|^{p+1}. \tag{3}$$

Доказательство следующей теоремы см. в Приложении 1 (литературный обзор см. в [11]).

Algorithm 1. Ускоренный Метаалгоритм (УМ) (УМ(x_0, f, g, p, H, k))

1: **Input:** $p\in\mathbb{N}$, $f\colon\mathbb{R}^d\to\mathbb{R}$, $g\colon\mathbb{R}^d\to\mathbb{R}$, $H>0$.

2: $A_0 = 0, y_0 = x_0$.

3: **for** $k = 0$ **to** $k = K-1$

4: Определить пару $\lambda_{k+1} > 0$ и $y_{k+1} \in \mathbb{R}^d$ из условий

$$\frac{1}{2} \le \lambda_{k+1} \frac{H\|y_{k+1} - \tilde{x}_k\|^{p-1}}{p!} \le \frac{p}{p+1},$$

где

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \widetilde{\Omega}^k(y) := \Omega_p(f, \tilde{x}_k; y) + g(y) + \frac{H}{(p+1)!}\|y - \tilde{x}_k\|^{p+1} \right\}, \tag{4}$$

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

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k.$$

5: $x_{k+1} := x_k - a_{k+1}\nabla f(y_{k+1}) - a_{k+1}\nabla g(y_{k+1})$.

6: **end for**

7: **return** y_K

Теорема 1. *Пусть y_k – выход алгоритма* 1 *УМ*(x_0, f, g, p, H, k) *после k итераций при $p \ge 1$ и $H \ge (p+1)L_{p,f}$. Тогда*

$$F(y_k) - F(x_*) \le \frac{c_p H R^{p+1}}{k^{\frac{3p+1}{2}}}, \tag{5}$$

где $c_p = 2^{p-1}(p+1)^{\frac{3p+1}{2}}/p!$, $R = \|x_0 - x^*\|$.

Более того, при $p \ge 2$ для достижения точности ε: $F(y_k) - F(x_*) \le \varepsilon$ на каждой итерации УМ вспомогательную задачу (4) придется перерешивать для подбора пары (λ_{k+1}, y_{k+1}) не более чем $O(\ln(\varepsilon^{-1}))$ раз.

Заметим, что приведенная выше теорема будет справедлива и при условии $H \ge 2L_{p,f}$ (независимо от $p \in \mathbb{N}$). Это выводится из (3). Условие $H \ge (p+1)L_{p,f}$ было использовано, поскольку оно гарантирует выпуклость вспомогательной подзадачи (4) (см. [17]). При этом условии и $g \equiv 0$ для $p = 1, 2, 3$ существуют эффективные способы решения вспомогательной задачи (4) (см. [17]). Для $p = 1$ существует явная формула для решения (4), для $p = 2, 3$ сложность (4) такая же (с точностью до логарифмического по ε множителя), как у итерации метода Ньютона (см. [17]).

Отметим, что вспомогательную задачу (4) не обязательно решать точно: достаточно (см. [11], [18]) найти точку $\tilde{y}_{k+1}$, удовлетворяющую условию

$$\left\|\nabla\widetilde{\Omega}^k(\tilde{y}_{k+1})\right\| \le \frac{1}{4p(p+1)}\|\nabla F(\tilde{y}_{k+1})\|. \tag{6}$$

Такая модификация приведет лишь к появлению множителя $12/5$ в правой части (5).

Будем говорить, что функция F является r-равномерно выпуклой ($p+1 \ge r \ge 2$) с константой $\sigma_r > 0$, если

$$F(y) \ge F(x) + \langle \nabla F(x), y - x\rangle + \frac{\sigma_r}{r}\|y - x\|^r, \quad x, y \in \mathbb{R}^d. \tag{7}$$

В этом случае, используя [19]:

$$F(\tilde{y}_{k+1}) - F(x_*) \le \frac{r-1}{r}\left(\frac{1}{\sigma_r}\right)^{\frac{1}{r-1}}\|\nabla F(\tilde{y}_{k+1})\|^{\frac{r}{r-1}},$$

можно завязать критерий (6) на желаемую точностью (по функции) решения исходной задачи ε (см. [11]): $\left\|\nabla\widetilde{\Omega}^{k}(\tilde{y}_{k+1})\right\| = O((\varepsilon^{r-1}\sigma_r)^{1/r})$.

Более того, для $p = 1$ приведенные здесь выкладки можно уточнить, подчеркнув тем самым, что сложность решения вспомогательной задачи может даже не зависеть от ε. Оказывается (см. [16]), что условие

$$\left\|\tilde{y}_{k+1} - y^*_{k+1}\right\| \le \frac{H}{3H + 2L_1^g}\left\|\tilde{x}_k - y^*_{k+1}\right\|, \tag{8}$$

где y^*_{k+1} – точное решение задачи (4), а L_1^g – константа Липшица градиента ∇g, в теоретическом плане гарантирует то же, что и условие (6) при $p = 1$. А именно, теорема 1 останется верной с добавлением в правую часть (5) множителя 12/5.

Отметим, что оценка скорости сходимости (5) с точностью до числового множителя c_p не может быть улучшена для класса выпуклых задач (1) с липшицевой p-й производной и для широкого класса тензорных методов порядка p, описанном в [17]. При дополнительном предположении равномерной выпуклости F оптимальный метод можно построить на базе УМ с помощью процедуры рестартов (см. [11]) (см. алгоритм 2).

Algorithm 2. Рестартованный УМ(x_0, f, g, p, r, σ_r, H, k)

1: **Input:** r-равномерно выпуклая функция $F = f + g : \mathbb{R}^d \to \mathbb{R}$ с константой σ_r и УМ(x_0, f, g, p, H, K).

2: $z_0 = x_0$.

3: **for** $k = 0$ **to** K

4: $R_k = R_0 \cdot 2^{-k}$,

$$N_k = \max\left\{\left\lceil\left(\frac{rc_pH2^r}{\sigma_r}R_k^{p+1-r}\right)^{\frac{2}{3p+1}}\right\rceil, 1\right\}. \tag{9}$$

5: $z_{k+1} := y_{N_k}$, где y_{N_k} – выход УМ(z_k, f, g, p, H, N_k).

6: **end for**

7: **return** z_K

Теорема 2. *Пусть y_k – выход алгоритма* 2 *после k итераций. Тогда если $H \ge (p+1)L_{p,f}$, $\sigma_r > 0$, то общее число вычислений* (4) *для достижения $F(y_k) - F(x_*) \le \varepsilon$ будет*:

$$N = \tilde{O}\left(\left(\frac{HR^{p+1-r}}{\sigma_r}\right)^{\frac{2}{3p+1}}\right),$$

где $\tilde{O}(\,)$ – означает то же самое, что $O(\,)$ с точностью до множителя $\ln(\varepsilon^{-1})$.

Все, что было сказано после теоремы 1, можно отметить и в данном случае.

3. ПРИЛОЖЕНИЯ

3.1. Ускоренные методы композитной оптимизации

Если не думать о сложности решения подзадачи (4), например, считать, что g – какая-то простая функция и (4) решается по явным формулам (как, например, для задачи LASSO), то УМ описывает класс ускоренных методов (1-, 2-, 3-го, ... порядков) композитной оптимизации (см. [1], [2], [6]). При этом функция g не обязана быть гладкой. В общем случае в строчке 5 алгоритма 1 под $\nabla g(y_{k+1})$ следует понимать такой субградиент функции g в точке y_{k+1}, с которым суб-

градиент правой части (4) равен (близок) к нулю (немного переписав метод, от последнего ограничения можно отказаться). Отметим, что при $p = 1$ необходимость в поиске параметра λ_{k+1} исчезает, что делает метод заметно проще.

3.2. Ускоренные проксимальные методы. Каталист

Если считать $p = 1$, а $f \equiv 0$, $H > 0$, то получится ускоренный проксимальный метод. Отличительная особенность такого метода (см. также [16]) от других известных ускоренных проксимальных методов заключается в том, что не требуется очень точно решать вспомогательную задачу. Критерий (8) и сильная (2-равномерная) выпуклость вспомогательной подзадачи (4) указывают на то, что сложность решения (8) может не зависеть от желаемой точности решения исходной задачи ε. Таким образом, не теряется логарифмический множитель при использовании такой проксимальной оболочки для ускорения различных неускоренных процедур. Собственно, последнее направление получило название Каталист (см. [4]). До настоящего момента идея (Каталист) использования ускоренной проксимальной оболочки для ‘обертывания’ неускоренных методов, решающих вспомогательную задачу (4) на каждой итерации (при должном выборе параметра H), являлась наиболее общей идеей разработки ускоренных методов для разных задач. Мы получаем Каталист просто как частный случай УМ. Примеры использования Каталист будут приведены в п. 3.4.

3.3. Разделение оракульных сложностей

Если считать, что для g имеем $L_{p,g} < \infty$ (см. (2)) и на вспомогательную задачу (4) смотреть как на равномерно выпуклую достаточно гладкую задачу (с $f := g$, $g(x) := \Omega_p\left(f, \tilde{x}_k; x\right) + \frac{(p+1)L_{p,f}}{(p+1)!}\|x - \tilde{x}_k\|^{p+1}$), то для решения (4), в свою очередь, можно использовать Рестартованный УМ с $H \simeq (p+1)L_{p,g}$. В случае, когда $L_{p,f} \le L_{p,g}$ удается получить такие оценки сложности (см. [11], [3]) (см. теорему 1):

$$N_f = \tilde{O}\left(\left(\frac{L_{p,f}R^{p+1}}{\varepsilon}\right)^{\frac{2}{3p+1}}\right) \text{ – число вызовов оракула для функции } f,$$

$$N_g = \tilde{O}\left(\left(\frac{L_{p,g}R^{p+1}}{\varepsilon}\right)^{\frac{2}{3p+1}}\right) \text{ – число вызовов оракула для функции } g.$$

Вызов оракула подразумевает вычисление (старших) производных до порядка p включительно. Таким образом, число вызовов оракула для каждой из функции f, g является квазиоптимальным, т.е. оптимальным с точностью до логарифмического (от желаемой точности по функции) множителя. Аналогичные оценки можно получить и в r-равномерно ($r \ge 2$) выпуклом случае (см. п. 3.4).

Заметим, что при $p = 1$ внутреннюю задачу (4) не обязательно решать Рестартованным УМ. Можно использовать (ускоренные) покомпонентные и безградиентные методы, методы редукции дисперсии (см. [1], [3], [20]). Причем ускорение можно получить из базовых неускоренных вариантов этих методов с помощью УМ (см. п. 3.2). По сравнению с оболочкой, использованной в [10], УМ дает оценку сложности на логарифмический множитель лучше. Это следует из теоретического анализа и было подтверждено в экспериментах (см. [21]).

3.4. Ускоренные методы для седловых задач

Следуя, например, [9], [12], рассмотрим выпукло-вогнутую седловую задачу

$$\min_{x\in\mathbb{R}^{d_x}}\{F(x) := f(x) + \underbrace{\max_{y\in\mathbb{R}^{d_y}}\{G(x,y) - h(y)\}}_{g(x)=G(x,y^*(x))-h(y^*(x))}\}, \tag{10}$$

где $y^*(x) = \operatorname{argmax}_{y\in\mathbb{R}^{d_y}}\{G(x,y) - h(y)\}$. Будем считать, что ∇f, ∇G, ∇h являются соответственно L_f, L_G, L_h-липшицевыми. Также будем считать, что $f(x) + G(x,y)$ является μ_x-сильно (2-равномерно) выпуклой по x, а $G(x,y) - h(y)$ является μ_y-сильно (2-равномерно) вогнутой по y. Тогда $F(x)$ будет μ_x-сильно выпуклой, а ∇g будет $L_g = (L_G + 2L_G^2/\mu_y)$-липшицевым (см. [9], [12]).

Если считать, что доступен ∇g, то внешнюю задачу (10) можно решать ускоренным слайдингом (например, в варианте УМ с $p = 1$, см. п. 3.3) за $\tilde{O}(\sqrt{L_f/\mu_x})$ вычислений ∇f и $\tilde{O}(\sqrt{L_g/\mu_x})$ вычислений ∇g.

Чтобы приближенно посчитать $\nabla g(x) = \nabla_x G(x, y^*(x))$, надо решить (с достаточной точностью) вспомогательную задачу в (10), т.е. найти с нужной точностью $y^*(x)$. Это, в свою очередь, также можно сделать с помощью слайдинга (УМ с $p = 1$) за $\tilde{O}(\sqrt{L_h/\mu_y})$ вычислений ∇h и $\tilde{O}(\sqrt{L_G/\mu_y})$ вычислений $\nabla_y G$.

Резюмируя написанное, получаем, что исходную задачу (10) можно решить за $\tilde{O}(\sqrt{L_f/\mu_x})$ вычислений ∇f, $\tilde{O}(\sqrt{L_g/\mu_x}) \simeq \tilde{O}\left(\sqrt{L_G^2/(\mu_x\mu_y)}\right)$ вычислений $\nabla_x G$, $\tilde{O}\left(\sqrt{L_G^3/(\mu_x\mu_y^2)}\right)$ вычислений $\nabla_y G$, $\tilde{O}\left(\sqrt{L_h L_G^2/(\mu_x\mu_y^2)}\right)$ вычислений ∇h. Поменяв порядок взятия min и max аналогичным образом, можно прийти к оценкам $\tilde{O}(\sqrt{L_h/\mu_y})$ вычислений ∇h, $\tilde{O}\left(\sqrt{L_G^2/(\mu_x\mu_y)}\right)$ вычислений $\nabla_y G$, $\tilde{O}\left(\sqrt{L_G^3/(\mu_x^2\mu_y)}\right)$ вычислений $\nabla_x G$, $\tilde{O}\left(\sqrt{L_f L_G^2/(\mu_x^2\mu_y)}\right)$ вычислений ∇f.

Оценки, полученные на число вычислений $\nabla_x G$ и ∇f, в последнем случае не являются оптимальными (см. [12]). Чтобы улучшить данные оценки (сделать их оптимальными с точностью до логарифмических множителей (см. [12])), воспользуемся Каталистом (см. п. 3.2) (УМ, с $p = 1$, $H \gg \mu_x$, $f \equiv 0$, $g = F$, где F определяется (10)). Если параметр метода H, то число итераций метода будет $\tilde{O}(\sqrt{H/\mu_x})$ (см. теорему 2). На каждой итерации необходимо будет решать с должной точностью задачу вида (10), в которой $L_f := L_f + H$, $\mu_x := \mu_x + H \simeq H$. Таким образом, для решения внутренней седловой задачи потребуется $\tilde{O}(\sqrt{L_h/\mu_y})$ вычислений ∇h, $\tilde{O}\left(\sqrt{L_G^2/(H\mu_y)}\right)$ вычислений $\nabla_y G$, $\tilde{O}\left(\sqrt{L_G^3/(H^2\mu_y)}\right)$ вычислений $\nabla_x G$, $\tilde{O}\left(\sqrt{(L_f + H)L_G^2/(H^2\mu_y)}\right)$ вычислений ∇f. Считая для наглядности $L_f \geq L_G$, выберем $H = L_G$. Тогда итоговые оценки на число вычислений соответствующих градиентов будут такие: $\tilde{O}(\sqrt{L_h L_G/(\mu_x\mu_y)})$ вычислений ∇h, $\tilde{O}\left(\sqrt{L_G^2/(\mu_x\mu_y)}\right)$ вычислений $\nabla_y G$, $\tilde{O}\left(\sqrt{L_G^2/(\mu_x\mu_y)}\right)$ вычислений $\nabla_x G$, $\tilde{O}\left(\sqrt{L_f L_G/(\mu_x\mu_y)}\right)$ вычислений ∇f.

За счет использования УМ приведенная выше схема улучшает похожую схему рассуждений из [12] на логарифмический (по желаемой точности решения задачи) множитель, и обобщает ее на случай отличных от тождественного нуля функций f и h.

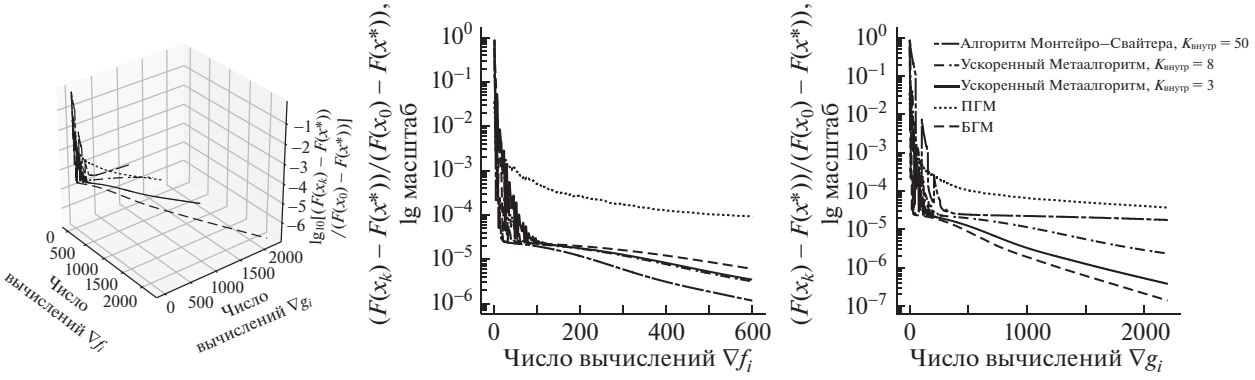
Приведенная здесь схема рассуждений наглядно демонстрирует, как из одной универсальной схемы ускорения удается получить (‘собрать как в конструкторе’) оптимальный метод с точностью до логарифмического (по желаемой точности множителя (при $f \equiv 0$ и $h \equiv 0$ – только при этих условиях известны нижние оценки (см. [12])).

3.5. Сравнение с алгоритмом Монтейро–Свайтера

Следуя [22], рассмотрим задачу оптимизации

$$\min_{x\in\mathbb{R}^n}\{F(x) := \underbrace{\log\left(\sum_{k=1}^{p}\exp(\langle A_k, x\rangle)\right)}_{=f(x)} + \underbrace{\frac{1}{2}\|Gx\|_2^2}_{=g(x)}\},$$

где $n = 500$, $p = 20\,000$, A – разреженная $p\times n$ матрица с коэффициентом разреженности 0.001 (под коэффициентом разреженности в данном случае понимается отношение числа ненулевых



Фиг. 1. Зависимость величины $(F(x_k) - F(x^*))/(F(x_0) - F(x^*))$ (в log масштабе) от числа вычислений компонент градиентов ∇f_i и ∇g_i. Двухмерные проекции.

элементов матрицы к общему числу ее элементов), чьи ненулевые элементы есть независимые одинаково распределенные случайные величины из равномерного распределения $\mathcal{U}(-1,1)$, а матрица G^2 получается из следующего выражения:

$$G^2 = \sum_{i=1}^{n} \lambda_i \tilde{e}_i^{\top} \tilde{e}_i,$$

где $\sum_{i=1}^{n} \lambda_i = 1$ и $[\tilde{e}_i]_j \sim \mathcal{U}(1,2)$ для каждой пары i, j.

Здесь f имеет липшицев градиент с константой Липшица:

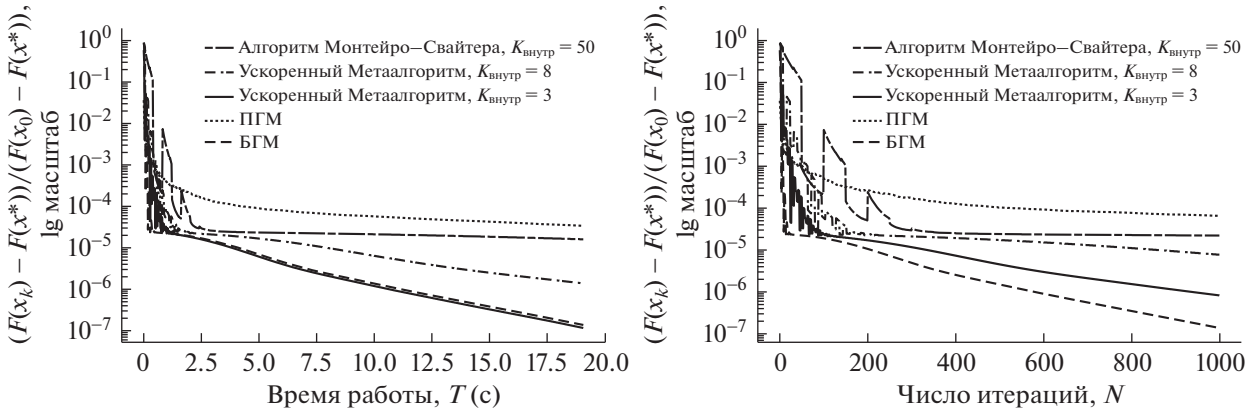
$$L_f = \max_{i=1,\ldots,n} \left\| A^{\langle k \rangle} \right\|_2^2,$$

где через $A^{\langle k \rangle}$ обозначен k-й столбец матрицы A.

На примере данной задачи сравним работу ускоренных методов, полученных с помощью алгоритма Монтейро–Свайтера (см. [7]) ($L = 20L_f$) и с помощью УМ ($H = L_f$) при использовании для решения вспомогательной задачи покомпонентного градиентного метода Нестерова (ПГМ) (см. [23]) ($\beta = 1/2$).

На фиг. 1 покомпонентный метод, ускоренный с помощью алгоритма Монтейро–Свайтера, сравнивается с методом, ускоренным оболочкой УМ с различным числом итераций метода для решений вспомогательной задачи ($K_{\text{внутр}} = k$ соответствует kn итерациям покомпонентного метода), и быстрым градиентным методом (БГМ). Представлен трехмерный график зависимости величины $(F(x_k) - F(x^*))/(F(x_0) - F(x^*))$ (в log масштабе, $F(x^*)$ выбирается равным значению F в точке, полученной после 25000 итераций БГМ) от числа вычислений компонент градиентов ∇f_i и ∇g_i, а также его двухмерные проекции. Так как некоторые методы требуют вычисления полного значения градиента (∇f или ∇g), обращения к оракулам в таком случае учитываются с весом $t_1/t_2 \approx 2.5$, где t_1 – среднее время вычисления полного градиента, t_2 – среднее время вычисления одной компоненты. Как можно видеть из графиков, число обращений к оракулу ∇f_i ускоренного с помощью оболочки УМ метода меньше, чем у быстрого градиентного метода. Кроме того, оболочка УМ позволяет значительно сократить число обращений к оракулу ∇g_i по сравнению с алгоритмом Монтейро–Свайтера.

На фиг. 2 сравнивается работа методов в зависимости от времени работы и числа итераций внутреннего метода. Как видно на фиг. 2, ускоренный с помощью оболочки УМ метод сходится по времени работы с большей скоростью, чем метод, ускоренный с помощью алгоритма Монтейро–Свайтера, а также с большей скоростью, чем быстрый градиентный метод.



Фиг. 2. Зависимость величины $(F(x_k) - F(x^*))/(F(x_0) - F(x^*))$ (в log масштабе) от времени работы и числа итераций внутреннего метода.

4. ВОЗМОЖНЫЕ ОБОБЩЕНИЯ

Приводимые выше конструкции существенным образом базируются на том, что рассматриваются задачи безусловной оптимизации и используется евклидова норма. На данный момент открытым остается вопрос о перенесении приведенных в статье результатов на задачи безусловной оптимизации с заменой евклидовой нормы на дивергенцию Брэгмана (см. [1], [5]). Тем более открытым остается вопрос об использовании других (более общих) моделей в построении мажоранты целевой функции (3) (см. [1]).

В [24] было подмечено (в том числе и в модельной общности), что для получения оптимальных версий ускоренных алгоритмов для задач стохастической оптимизации нужно уметь оценивать, как накапливается малый шум в градиенте в таких методах. Таким образом, строятся ускоренные стохастические градиентные методы на базе ускоренных не стохастических (детерминированных) и конструкции, названной минибатчингом (замена градиента в детерминированном методе его оценкой, построенной на базе стохастических градиентов). Насколько нам известно, для тензорных методов вопрос построения ускоренных методов для задач стохастической оптимизации остается открытым. В частности, не известен ответ на такой вопрос: верно ли, что для задач сильно выпуклой стохастической оптимизации требования к точности аппроксимации старших производных с помощью минибатчинга снижаются по мере роста порядка производных, как это имеет место в невыпуклом случае (см. [25])? Для ответа на этот вопрос для тензорных методов также как и для градиентных $p = 1$ может пригодиться анализ чувствительности исследуемых методов к неточности в вычислении производных. Некоторый задел в этом направлении уже имеется (см. [26]). В частности, при $p = 1$ УМ демонстрирует стандартное для ускоренных методов накопление неточностей в градиенте (см. [24]).

Из статьи может показаться, что для безусловных достаточно гладких задач выпуклой оптимизации предлагаемый в статье подход дает возможность всегда строить “оптимальные” методы. На самом деле это не совсем так. Во-первых, построение оптимальных методов даже на базе одного только УМ может быть совсем не простой задачей, как показывает пример из п. 3.4. Во-вторых, оговорка “с точностью до логарифмических множителей” весьма существенна. В частности, до сих пор остается открытым вопрос о том, устраним ли логарифмический мультипликативный зазор (по желаемой точности решения задачи по функции) между нижними оценками и тем, что дает УМ и другие ускоренные тензорные методы ($p \geq 2$) (см. теорему 1). В-третьих, упомянутые нижние оценки были получены для класса крыловских методов (для тензорных методов чуть иначе (см. [17])), однако, предлагаемая оболочка УМ в некоторых вариантах ее использования, в том числе в проксимальном варианте (Катализ) (см. п. 3.2), выводит из класса допустимых методов, для которого были полученые нижние оценки.

ПРИЛОЖЕНИЕ 1

В этом приложении представлено доказательство теоремы 1, основанное на доказательстве из [14], с учетом добавления композитной функции. Следующая теорема базируется на теореме 2.1 из [14].

Теорема 3. *Пусть $(y_k)_{k\geq 1}$ – это последовательность точек в $\mathbb{R}^d$, и $(\lambda_k)_{k\geq 1}$ – это последовательность в $\mathbb{R}_+$. Определим $(a_k)_{k\geq 1}$ такой, что $\lambda_k A_k = a_k^2$ и $A_k = \sum_{i=1}^k a_i$. Для любого $k \geq 0$ определим*

$$x_k = x_0 - \sum_{i=1}^k a_i(\nabla f(y_i) + g'(y_i)) \quad и \quad \tilde{x}_k := \frac{a_{k+1}}{A_{k+1}} x_k + \frac{A_k}{A_{k+1}} y_k.$$

Также предположим, что если для некоторого $\sigma \in [0,1]$ имеем

$$\|y_{k+1} - (\tilde{x}_k - \lambda_{k+1}\nabla f(y_{k+1}))\| \leq \sigma \|y_{k+1} - \tilde{x}_k\|, \tag{11}$$

тогда для любого $x \in \mathbb{R}^d$ верны неравенства

$$F(y_k) - F(x) \leq \frac{2\|x\|^2}{\left(\sum_{i=1}^k \sqrt{\lambda_i}\right)^2},$$

и

$$\sum_{i=1}^k \frac{A_i}{\lambda_i}\|y_i - \tilde{x}_{i-1}\|^2 \leq \frac{\|x^*\|^2}{1-\sigma^2}.$$

Для доказательства этой теоремы мы введем дополнительные леммы, основанные на леммах 2.2–2.5 и 3.1 из [14], леммы 2.6 и 3.3 могут использоваться без изменений.

Лемма 1. *Пусть $\psi_0(x) = \frac{1}{2}\|x - x_0\|^2$, и по индукции определим $\psi_k(x) = \psi_{k-1}(x) + a_k\Omega_1(F, y_k, x)$, тогда $x_k = x_0 - \sum_{i=1}^k a_i(\nabla f(y_i) + g'(y_i))$ – это минимизатор функции ψ_k, и верно $\psi_k(x) \leq A_k F(x) + \frac{1}{2}\|x - x_0\|^2$, где $A_k = \sum_{i=1}^k a_i$.*

Лемма 2. *Пусть z_k такая, что*

$$\psi_k(x_k) - A_k F(z_k) \geq 0.$$

Тогда для любого x имеем

$$F(z_k) \leq F(x) + \frac{\|x - x_0\|^2}{2A_k}.$$

Доказательство. Из леммы 1 можно получить, что

$$A_k F(z_k) \leq \psi_k(x_k) \leq \psi_k(x) \leq A_k F(x) + \frac{1}{2}\|x - x_0\|^2.$$

Лемма 3. *Для любого x верно следующее неравенство*:

$$\psi_{k+1}(x) - A_{k+1}F(y_{k+1}) - (\psi_k(x_k) - A_k F(z_k)) \geq$$
$$\geq A_{k+1}(\nabla f(y_{k+1}) + g'(y_{k+1}))\left(\frac{a_{k+1}}{A_{k+1}}x + \frac{A_k}{A_{k+1}}z_k - y_{k+1}\right) + \frac{1}{2}\|x - x_k\|^2.$$

Доказательство. Во-первых, простыми вычислениями получим

$$\psi_k(x) = \psi_k(x_k) + \frac{1}{2}\|x - x_k\|^2,$$

и

$$\psi_{k+1}(x) = \psi_k(x_k) + \frac{1}{2}\|x - x_k\|^2 + a_{k+1}\Omega_1(f, y_{k+1}, x),$$

таким образом имеем

$$\psi_{k+1}(x) - \psi_k(x_k) = a_{k+1}\Omega_1(F, y_{k+1}, x) + \frac{1}{2}\|x - x_k\|^2. \tag{12}$$

Теперь мы хотим, чтобы $A_{k+1}F(z_{k+1}) - A_kF(z_k)$ было нижней оценкой неравенства (12), когда вычисляем $x = x_{k+1}$. Используя $\Omega_1(F, y_{k+1}, z_k) \le f(z_k)$, мы получаем

$$a_{k+1}\Omega_1(F, y_{k+1}, x) = A_{k+1}\Omega_1(F, y_{k+1}, x) - A_k\Omega_1(F, y_{k+1}, x) = A_{k+1}\Omega_1(F, y_{k+1}, x) - A_k\nabla F(y_{k+1})(x - z_k) -$$
$$- A_k\Omega_1(F, y_{k+1}, z_k) = A_{k+1}\Omega_1\left(F, y_{k+1}, x - \frac{A_k}{A_{k+1}}(x - z_k)\right) - A_k\Omega_1(F, y_{k+1}, z_k) \ge A_{k+1}F(y_{k+1}) - A_kF(z_k) +$$
$$+ A_{k+1}(\nabla f(y_{k+1}) + g'(y_{k+1}))\left(\frac{a_{k+1}}{A_{k+1}}x + \frac{A_k}{A_{k+1}}z_k - y_{k+1}\right),$$

что завершает доказательство.

Лемма 4. *Пусть* $\lambda_{k+1} := \dfrac{a_{k+1}^2}{A_{k+1}}$ *и* $\tilde{x}_k := \dfrac{a_{k+1}}{A_{k+1}}x_k + \dfrac{A_k}{A_{k+1}}y_k$, *тогда имеем*

$$\psi_{k+1}(x_{k+1}) - A_{k+1}F(y_{k+1}) - (\psi_k(x_k) - A_kF(y_k)) \ge$$
$$\ge \frac{A_{k+1}}{2\lambda_{k+1}}(\|y_{k+1} - \tilde{x}_k\|^2 - \|y_{k+1} - (\tilde{x}_k - \lambda_{k+1}(\nabla f(y_{k+1})) + g'(y_{k+1}))\|^2).$$

А применив дополнительно неравенство (11), *получим*

$$\psi_k(x_k) - A_kF(y_k) \ge \frac{1 - \sigma^2}{2}\sum_{i=1}^{k}\frac{A_i}{\lambda_i}\|y_i - \tilde{x}_{i-1}\|^2.$$

Доказательство. Используем лемму 3 при $z_k = y_k$ и $x = x_{k+1}$ и получаем, что (при $\tilde{x} := \dfrac{a_{k+1}}{A_{k+1}}x + \dfrac{A_k}{A_{k+1}}y_k$)

$$(\nabla f(y_{k+1}) + g'(y_{k+1}))\left(\frac{a_{k+1}}{A_{k+1}}x + \frac{A_k}{A_{k+1}}y_k - y_{k+1}\right) + \frac{1}{2A_{k+1}}\|x - x_k\|^2 = (\nabla f(y_{k+1}) + g'(y_{k+1}))(\tilde{x} - y_{k+1}) +$$
$$+ \frac{1}{2A_{k+1}}\left\|\frac{A_{k+1}}{a_{k+1}}\left(\tilde{x} - \frac{A_k}{A_{k+1}}y_k\right) - x_k\right\|^2 = (\nabla f(y_{k+1}) + g'(y_{k+1}))(\tilde{x} - y_{k+1}) + \frac{A_{k+1}}{2a_{k+1}^2}\left\|\tilde{x} - \left(\frac{a_{k+1}}{A_k}x_k + \frac{A_k}{A_{k+1}}y_k\right)\right\|^2.$$

Откуда следует неравенство

$$\psi_{k+1}(x_{k+1}) - A_{k+1}F(y_{k+1}) - (\psi_k(x_k) - A_kF(y_k)) \ge$$
$$\ge A_{k+1}\min_{x\in\mathbb{R}^d}\left\{(\nabla f(y_{k+1}) + g'(y_{k+1}))(x - y_{k+1}) + \frac{1}{2\lambda_{k+1}}\|x - \tilde{x}_k\|^2\right\}.$$

Значение минимума можно легко посчитать.

Для первого выражения теоремы 3 достаточно объединить лемму 4 с леммой 2 и леммой 2.5 из [14]. Второе выражение в теореме 3 следует из леммы 4 и леммы 1.

Следующая лемма доказывает, что минимизация ряда Тэйлора порядка p для (4) может быть представлена как неявный градиентный шаг для некоторого большого размера шага.

Лемма 5. *Неравенство* (11) *верно при* $\sigma = 1/2$ *для* (4), *из этого следует, что*

$$\frac{1}{2} \le \lambda_{k+1}\frac{L_p\|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \le \frac{p}{p+1}. \tag{13}$$

Доказательство. Из условия оптимальности имеем

$$\nabla_y f_p(y_{k+1}, \tilde{x}_k) + \frac{L_p(p+1)}{p!}(y_{k+1} - \tilde{x}_k)\|y_{k+1} - \tilde{x}_k\|^{p-1} + g'(y_{k+1}) = 0. \tag{14}$$

Откуда следует, что

$$y_{k+1} - (\tilde{x}_k - \lambda_{k+1}(\nabla f(y_{k+1}) + g'(y_{k+1}))) = \lambda_{k+1}(\nabla f(y_{k+1}) + g'(y_{k+1})) - \\ - \frac{p!}{L_p(p+1)\|y_{k+1} - \tilde{x}_k\|^{p-1}}(\nabla_y f_p(y_{k+1}, \tilde{x}_k) + g'(y_{k+1})).$$

Используя ряд Тэйлора, для градиента функции получаем

$$\|\nabla f(y) - \nabla_y f_p(y, x)\| \le \frac{L_p}{p!}\|y - x\|^p,$$

таким образом верно

$$\|y_{k+1} - (\tilde{x}_k - \lambda_{k+1}(\nabla f(y_{k+1}) + g'(y_{k+1})))\| \le \lambda_{k+1}\frac{L_p}{p!}\|y_{k+1} - \tilde{x}_k\|^p + \\ + \left|\lambda_{k+1} - \frac{p!}{L_p(p+1)\|y_{k+1} - \tilde{x}_k\|^{p-1}}\right|\|\nabla_y f_p(y_{k+1}, \tilde{x}_k) + g'(y_{k+1})\| \le \\ \le \|y_{k+1} - \tilde{x}_k\|\left(\lambda_{k+1}\frac{L_p}{p!}\|y_{k+1} - \tilde{x}_k\|^{p-1} + \left|\lambda_{k+1}\frac{L_p(p+1)\|y_{k+1} - \tilde{x}_k\|^{p-1}}{p!} - 1\right|\right) = \\ = \|y_{k+1} - \tilde{x}_k\|\left(\frac{\eta}{p} + \left|\eta\frac{p+1}{p} - 1\right|\right),$$

где мы используем (14) во втором неравенстве, и предполагаем $\eta := \lambda_{k+1}\dfrac{L_p\|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!}$ в последнем равенстве. Итоговый результат получаем из предположения, что $1/2 \le \eta \le p/(p+1)$ в (13).

В заключение, если мы заменим $\|x^*\|$ на $\|x_0 - x^*\|$ в лемме 3.3 и используем лемму 3.4 из [14], то получим доказательство теоремы 1.

ПРИЛОЖЕНИЕ 2

Докажем теорему 2.

Доказательство. Так как функция F является r-равномерно выпуклой, то мы получаем

$$R_{k+1} = \|z_{k+1} - x_*\| \le \left(\frac{r(F(z_{k+1}) - F(x_*))}{\sigma_r}\right)^{1/r} \overset{(5)}{\le} \left(\frac{r\left(\dfrac{c_p L_p R_k^{p+1}}{N_k^{\frac{3p+1}{2}}}\right)}{\sigma_r}\right)^{1/r} = \left(\frac{rc_p L_p R_k^{p+1}}{\sigma_r N_k^{\frac{3p+1}{2}}}\right)^{1/r} \overset{(9)}{\le} \left(\frac{R_k^{p+1}}{2^r R_k^{p+1-r}}\right)^{1/r} = \frac{R_k}{2}.$$

Теперь вычислим общее число шагов метода 1:

$$\sum_{k=0}^{K} N_k \le \sum_{k=0}^{K}\left(\frac{rc_p L_p 2^r}{\sigma_r}R_k^{p+1-r}\right)^{\frac{2}{3p+1}} + K = \sum_{k=0}^{K}\left(\frac{rc_p L_p 2^r}{\sigma_r}(R_0 2^{-k})^{p+1-r}\right)^{\frac{2}{3p+1}} + K = \\ = \left(\frac{rc_p L_p 2^r R_0^{p+1-r}}{\sigma_r}\right)^{\frac{2}{3p+1}}\sum_{k=0}^{K} 2^{\frac{-2(p+1-r)k}{3p+1}} + K.$$

СПИСОК ЛИТЕРАТУРЫ

1. *Гасников А.В.* Современные численные методы оптимизации. Метод универсального градиентного спуска. М.: МФТИ, 2018.
2. *Nesterov Yu.* Lectures on convex optimization. V. 137. Berlin, Germany: Springer, 2018.

3. *Lan G.* Lectures on optimization. Methods for Machine Learning // https://pwp.gatech.edu/guanghui-lan/publications/

4. *Lin H., Mairal J., Harchaoui Z.* Catalyst acceleration for first-order convex optimization: from theory to practice // J. Machine Learning Res. 2017. V. 18. No. 1. P. 7854–7907.

5. *Doikov N., Nesterov Yu.* Contracting proximal methods for smooth convex optimization // arXiv:1912.07972.

6. *Gasnikov A., Dvurechensky P., Gorbunov E., Vorontsova E., Selikhanovych D., Uribe C.A., Jiang B., Wang H., Zhang S., Bubeck S., Jiang Q.* Near Optimal Methods for Minimizing Convex Functions with Lipschitz p-th Derivatives // Proceed. Thirty-Second Conf. Learning Theory. 2019. P. 1392–1393.

7. *Monteiro R.D.C., Svaiter B.F.* An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods // SIAM J. Optimizat. 2013. V. 23. № 2. P. 1092–1125.

8. *Nesterov Yu.* Inexact Accelerated High-Order Proximal-Point Methods // CORE Discussion paper 2020/8.

9. *Alkousa M., Dvinskikh D., Stonyakin F., Gasnikov A.* Accelerated methods for composite non-bilinear saddle point problem // arXiv:1906.03620.

10. *Ivanova A., Gasnikov A., Dvurechensky P., Dvinskikh D., Tyurin A., Vorontsova E., Pasechnyuk D.* Oracle Complexity Separation in Convex Optimization // arXiv:2002.02706

11. *Kamzolov D., Gasnikov A., Dvurechensky P.* On the optimal combination of tensor optimization methods // arXiv:2002.01004

12. *Lin T., Jin C., Jordan M.* Near-optimal algorithms for minimax optimization // arXiv:2002.02417.

13. *Gasnikov A., Dvurechensky P., Gorbunov E., Vorontsova E., Selikhanovych D., Uribe C.A.* Optimal Tensor Methods in Smooth Convex and Uniformly ConvexOptimization // Proc. Thirty-Second Conf. Learning Theory. 2019. P. 1374–1391.

14. *Bubeck S., Jiang Q., Lee Y.T., Li Y., Sidford A.* Near-optimal method for highly smooth convex optimization // Proc. Thirty-Second Conf. Learning Theory. 2019. P. 492–507.

15. *Jiang B., Wang H., Zhang S.* An optimal high-order tensor method for convex optimization // Proc. Thirty-Second Conf. Learning Theory. 2019. P. 1799–1801.

16. *Ivanova A., Grishchenko D., Gasnikov A., Shulgin E.* Adaptive Catalyst for smooth convex optimization // arXiv:1911.11271

17. *Nesterov Yu.* Implementable tensor methods in unconstrained convex optimization // Math. Program. 2019. P. 1–27.

18. *Kamzolov D., Gasnikov A.* Near-Optimal Hyperfast Second-Order Method for convex optimization and its Sliding // arXiv:2002.09050

19. *Grapiglia G.N., Nesterov Yu.* On inexact solution of auxiliary problems in tensor methods for convex optimization // arXiv:1907.13023

20. *Dvurechensky P., Gasnikov A., Tiurin A.* Randomized Similar Triangles Method: A unifying framework for accelerated randomized optimization methods (Coordinate Descent, Directional Search, Derivative-Free Method) // arXiv:1707.08486

21. Ссылка: исходный код экспериментов на GitHub https://github.com/dmivilensky/composite-accelerated-method

22. *Spokoiny V., Panov M.* Accuracy of Gaussian approximation in nonparametric Bernstein–von Mises Theorem // arXiv preprint arXiv:1910.06028. 2019.

23. *Nesterov Yu., Stich S.U.* Efficiency of the accelerated coordinate descent method on structured optimization problems // SIAM Journal on Optimization. 2017. Т. 27. №. 1. С. 110–123.

24. *Dvinskikh D., Tyurin A., Gasnikov A., Omelchenko S.* Accelerated and nonaccelerated stochastic gradient descent with model conception // arXiv:2001.03443

25. *Lucchi A., Kohler J.* A Stochastic Tensor Method for Non-convex Optimization // arXiv:1911.10367

26. *Baes M.* Estimate sequence methods: extensions and approximations // Inst.r Operat. Res. ETH, Zürich, Switzerland, 2009.