Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

On the optimal combination of tensor optimization methods

Dmitry Kamzolov[1], Alexander Gasnikov[1,2,3], Pavel Dvurechensky[4]

submitted: April 2, 2020

[1] Moscow Institute of Physics and Technology
Dolgoprudny, Russia
E-Mail: kamzolov.dmitry@phystech.edu
gasnikov@yandex.ru

[2] Institute for Information Transmission Problems
Moscow, Russia
E-Mail: gasnikov@yandex.ru

[3] Higher School of Economics
Moscow, Russia
E-Mail: gasnikov@yandex.ru

[4] Weierstrass Institute
Mohrenstr. 39
10117 Berlin, Germany
E-Mail: pavel.dvurechensky@wias-berlin.de

No. 2710
Berlin 2020

2010 *Mathematics Subject Classification.* 90C30, 90C25, 68Q25, 65K15.

Key words and phrases. Tensor methods, sliding, uniformly convex function, inexactness, Taylor expansion, complexity.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

On the optimal combination of tensor optimization methods

Dmitry Kamzolov, Alexander Gasnikov, Pavel Dvurechensky

Abstract

We consider the minimization problem of a sum of a number of functions having Lipshitz p-th order derivatives with different Lipschitz constants. In this case, to accelerate optimization, we propose a general framework allowing to obtain near-optimal oracle complexity for each function in the sum separately, meaning, in particular, that the oracle for a function with lower Lipschitz constant is called a smaller number of times. As a building block, we extend the current theory of tensor methods and show how to generalize near-optimal tensor methods to work with inexact tensor step. Further, we investigate the situation when the functions in the sum have Lipschitz derivatives of a different order. For this situation, we propose a generic way to separate the oracle complexity between the parts of the sum. Our method is not optimal, which leads to an open problem of the optimal combination of oracles of a different order.

1 Introduction

Higher-order (tensor) methods, which use the derivatives of the objective up to order p, recently have become an area of intensive research effort in optimization, despite the idea is quite old and goes back to the works of P. Chebyshev and L. Kantorovich ([5] and [18]). One of the reasons is that the lower complexity bounds were obtained in [2, 1, 26], which opened a question of optimal methods, and it was shown in [26] that Taylor expansion of a convex function can be made convex by appropriate regularization, leading to tractable tensor step implementable in practice. Recently nearly optimal methods were obtained in [26, 13], and extensions for Hölder continuous higher-order derivatives were proposed in [16, 28]. In this paper, we consider an interesting question that is still open in the theory of tensor methods. *Namely, if a tensor method minimizes a function f up to accuracy ε in $N_f(\varepsilon)$ oracle calls and possibly another tensor method minimizes a function g in $N_g(\varepsilon)$ oracle calls, is it possible to combine these two methods to minimize $f + g$ up to accuracy ε in $\tilde{O}(N_f(\varepsilon))$ oracle calls for f and $\tilde{O}(N_g(\varepsilon))$ oracle calls for g?* To say more, we would like to have a generic approach which can take as an input different particular algorithms for each component. For simplicity, we consider a sum of two functions, but we believe that the approach can be generalized for an arbitrary number of functions. Note that in the last few years, the answer to this question plays a crucial role in the development of optimal algorithms for convex decentralized distributed optimization [21, 20, 9, 14, 3, 27].

Some results in this direction are known for the first-order methods $p = 1$ [19, 22, 20, 3, 10] and for the case of the sum of two functions with the second being so simple that it can be incorporated directly in the tensor step [17] like in composite first-order methods [24]. Yet, the general theory on how to combine different methods to obtain optimal complexity for tensor methods is not yet developed for $p \geq 2$.

First, we consider uniformly convex sum of two functions $f + g$ each having Lipschitz derivatives of the same order p. Our approach is based on the recent framework of near-optimal tensor methods [13], which extends the algorithm of [23] to tensor methods. Our idea is to apply the near-optimal tensor

DOI 10.20347/WIAS.PREPRINT.2710 Berlin 2020

method to the sum, considering g as a composite and including it into the tensor step without its Taylor approximation. Then each tensor step requires to solve properly regularized uniformly convex auxiliary problem. This is again done by the nearly optimal tensor method. Since the auxiliary problem turns out to be very well conditioned, it is possible to solve it very fast, and we only need to call the oracle for g. The careful analysis allows to separate the oracle complexity as we call the oracle for f only on outer iterations and oracle for g only on the inner, resulting in the optimal number of oracle calls for f and for g separately. As a building block, we explain how to extend near-optimal tensor methods to work with inexact tensor step, extending the current theory since existing near-optimal methods assume that the tensor step is exact. If the function is not uniformly convex, one can use a standard regularization technique with a small regularization parameter.

Note, there exist number of accelerated envelopes that allows to accelerate tensor methods: Monteiro–Svaiter envelop [23, 25, 12, 17, 4], Doikov–Nesterov envelope [7]. Further we will use Monteiro–Svaiter envelope. Note that it seems that Doikov–Nesterov envelop and standard direct Nesterov's tensor acceleration [26] doesn't well suited for our purposes. Note also, that for all envelops for the moment it's not known with what accuracy we should solve auxiliary problem. In Monteiro–Svaiter envelop we working on this in Appendix B. Among different variants of Monteiro–Svaiter envelop we preferred variant from [4], but we generalize (see Appendixes) [4] on composite case [17] and on uniformly convex problem target functions [12].

Second, we consider the case when f and g has Lipshitz derivatives of different order p_f and p_g respectively. We apply a similar technique as above, but using non-accelerated tensor methods as building blocks. We demonstrate that in this case, complexities can also be separated, but they turn out to be not optimal. This states an open problem of an optimal combination of optimal methods that use oracles of a different order. As far as we know for the moment there exists only one optimal result concerns the methods of different orders. This is the result from [3], where authors considered sliding of optimal 0-order and 1-order methods.

2 Problem Statement and Preliminaries

In what follows, we work in a finite-dimensional linear vector space E. Its dual space, the space of all linear functions on E, is denoted by E^*. For $x \in E$ and $s \in E^*$, we denote by $\langle s, x\rangle$ the value of a linear function s at x. For the (primal) space E, we introduce a norm $\|\cdot\|_E$. Then the dual norm is defined in the standard way:

$$\|s\|_{E^*} = \max_{x\in E} \{\langle s, x\rangle : \|x\|_E \leq 1\}.$$

Finally, for a convex function f : **dom** $f \to R$ with **dom** $f \subseteq E$ we denote by $\nabla f(x) \in E^*$ one of its subgradients.

We consider the following convex optimization problem:

$$\min_{x\in E} F(x) = f(x) + g(x), \tag{1}$$

where $f(x)$ and $g(x)$ are convex functions with Lipschitz p-th derivative, it means that

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

Then Taylor approximation of function $f(x)$ can be written as follows:

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

By (2) and the standard integration we can get next inequality

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

Now we introduce an additional condition for the functions.

Definition 1. Function $F(x)$ is r-uniformly convex ($r \geq 2$) if

$$F(y) \geq F(x) + \langle \nabla F(x), y-x\rangle + \frac{\sigma_r}{r}\|y-x\|^r, \quad \forall x,y \in E$$

with constant σ_r

One of the main examples of r-uniformly convex functions is $\frac{1}{r}\|x\|^r$ from Lemma 5 [7].

Lemma 1. *For fixed $r \geq 2$, consider the following function:*

$$f_r(x) = \frac{1}{r}\|x\|^r, \quad x \in \mathbb{E}.$$

Function $f_r(x)$ is uniformly convex of degree r with $\sigma_r = 2^{2-r}$.

Problem (1) can be solved by tensor methods [26] or its accelerated versions [25], [4], [17], [13] . This methods have next basic step:

$$T_H(x) = \operatorname*{argmin}_y \left\{ \Omega_p(f+g,x;y) + \frac{H_p}{p!}\|y-x\|^{p+1} \right\}.$$

For $H_p \geq L_p$ this subproblem is convex and hence implementable. Note that this method does not use information about sum type problem and compute their derivatives the same number of times. We want to separate computation complexity of high-order derivatives for sum of two functions. In next section we will describe this idea in more details.

As an accelerated optimal method, we introduce Accelerated Taylor Descent (ATD) from [4]. But for our paper we need to get a composite variant of ATD.

Algorithm 1 is a generalization of ATD from [4] for composite optimization problem. It means, that we try to minimize sum of two functions $F(x) = f(x) + g(x)$, where $g(x)$ is a proper closed convex function and subproblem (4) with $g(x)$ is easy to solve. Note that if $g(x)$ smooth and has a gradient, so $g'(y_{k+1}) = \nabla g(y_{k+1})$, but if $g(x)$ has only subgradient, we should introduce $g'(y_{k+1})$. Similarly to (2.9) from [6] by using optimality condition for (4) we define

$$g'(y_{k+1}) = -\nabla\Omega_p(f,\tilde{x}_k;y_{k+1}) - \frac{(p+1)H_{p,f}}{p!}\|y_{k+1}-\tilde{x}_k\|^{p-1}(y_{k+1}-\tilde{x}_k)$$

Theorem 2. *Let $F(x) = f(x) + g(x)$, where f denote a convex function whose p^{th} derivative is L_p-Lipschitz, $g(x)$ is a proper closed convex function and let x_* denote a minimizer of F. Then CATD satisfies, with $c_p = 2^{p-1}(p+1)^{\frac{3p+1}{2}}/(p-1)!$,*

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

Algorithm 1 Composite Accelerated Taylor Descent

Input: convex function $f : \mathbb{R}^d \to \mathbb{R}$ such that $\nabla^p f$ is L_p-Lipschitz, proper closed convex $g : \mathbb{R}^d \to \mathbb{R}$.
Set $A_0 = 0, x_0 = y_0$
for $k = 0$ **to** $k = K - 1$ **do**
  Compute a pair $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ such that

$$\frac{1}{2} \leq \lambda_{k+1} \frac{H_{p,f} \cdot \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1},$$

  where

$$y_{k+1} = \operatorname*{argmin}_y \left\{ \Omega_p(f, \tilde{x}_k; y) + \frac{H_{p,f}}{p!} \|y - \tilde{x}_k\|^{p+1} + g(y) \right\}, \tag{4}$$

  and

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}, A_{k+1} = A_k + a_{k+1}, \text{ and } \tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k.$$

  Update $x_{k+1} := x_k - a_{k+1} \nabla f(y_{k+1}) - a_{k+1} g'(y_{k+1})$
end for
return y_K

where

$$R = \|x_0 - x^*\| \tag{6}$$

is the maximal radius of the initial set. Furthermore each iteration of ATD can be implemented in $\tilde{O}(1)$ calls to a p^{th}-order Taylor expansion oracle, where $\tilde{O}$ means up to logarithmic factors.

We prove this theorem similarly to the proof of [4] in Appendix A.

Now we assume that function $F(x)$ is additionally r-uniformly convex, hence we may get a speed up by using restarts. We formulate method and theorem for CATD with restarts.

Algorithm 2 CATD with restarts

Input: r-unformly convex function $F : \mathbb{R}^d \to \mathbb{R}$ with constant σ_r and CATD conditions.
Set $z_0 = x_0 = 0$ and $R_0 = \|z_0 - x_*\|$.
for $k = 0,$ **to** K **do**
  Set $R_k = R_0 \cdot 2^{-k}$ and

$$N_k = \max \left\{ \left\lceil \left(\frac{r c_p L_p 2^r}{\sigma_r} R_k^{p+1-r} \right)^{\frac{2}{3p+1}} \right\rceil, 1 \right\}. \tag{7}$$

  Set $z_{k+1} := y_{N_k}$ as the output of CATD started from z_k and run for N_k steps.
end for
return z_K

Theorem 3. *CATD with restarts for r-uniformly convex function F with constant σ_r converges with N_r steps of CATD per restart and with N_F total number of CATD steps, where*

$$N_F = \tilde{O}\left[\left(\frac{L_{p,f}R^{p+1-r}}{\sigma_r}\right)^{\frac{2}{3p+1}}\right].$$

We prove this theorem similarly to [12] in Appendix C.

3 Uniformly convex functions

We consider similar to (1) problem.

$$\min F(x) = f(x) + g(x), \tag{8}$$

where additionally $F(x)$ is r-uniformly convex function. We also assume, that $p + 1 \geq r$.

If we will use Algorithm 2 for problem (8) we get next convergence speed. To reach $F(x_N) - F(x^*) \leq \varepsilon$, we need $N_f + N_g$ iterations, where

$$N_f = \tilde{O}\left[\left(\frac{L_{p,f}R^{p+1-r}}{\sigma_r}\right)^{\frac{2}{3p+1}}\right], \tag{9}$$

$$N_g = \tilde{O}\left[\left(\frac{L_{p,g}R^{p+1-r}}{\sigma_r}\right)^{\frac{2}{3p+1}}\right]. \tag{10}$$

Note that for this method we compute $N_f + N_g$ derivatives for both $f(x)$ and $g(x)$ functions. We want to separate this computations and compute N_f derivatives for the function f and N_g derivatives for the function g.

Next we will describe the our framework. We assume that $L_{p,f} < L_{p,g}$, it means that $N_f < N_g$. For that case we consider problem 8 as a composite problem with $g(x)$ as a composite part. We solve this problem by Algorithm 2. In this algorithm we have tensor subproblem (4). To solve this subproblem we run another Algorithm 2 with objective function $\Omega_p(f, \tilde{x}_k; y) + \frac{H_{p,f}}{p!}\|y - \tilde{x}_k\|^{p+1} + g(y)$ up to the desired accuracy. As we will prove next, this subproblem may be solved linearly by the desired accuracy, so we should not worry too much about the level of the desired accuracy. We write more details about the correctness of this part and the more precise level of desired accuracy in Appendix B. As a result we get Algorithm 3.

Now we prove that this framework split computation's complexities.

Theorem 4. *Assume $F(x)$ is r-uniformly convex function ($r \geq 2$), $f(x)$ and $g(x)$ are convex functions with Lipshitz p-th derivative ($p \geq 1$, $p + 1 \geq r$) and $L_{p,f} < L_{p,g}$. Then by using our framework with $H_{p,f} = 2L_{p,f}$, method converges to $F(x_N) - F(x^*) \leq \varepsilon$ with N_f as* (9) *computations of derivatives $f(x)$ and N_g as* (10) *computation of derivatives $g(x)$.*

Proof. As we prove in 3 for the outer composite method with constant $H_{p,f} = 2L_{p,f}$ we need to make

$$N_{out} = \tilde{O}\left[\left(\frac{2pL_{p,f}R^{p+1-r}}{\sigma_r}\right)^{\frac{2}{3p+1}}\right]$$

Algorithm 3 Tensor Methods Combination

Input: r-unformly convex function $F(x) = f(x) + g(x)$ with constant σ_r, convex functions $f(x)$ and $g(x)$ such that $\nabla^p f$ is $L_{p,f}$-Lipschitz and $\nabla^p g$ is $L_{p,g}$-Lipschitz.
Set $z_0 = y_0 = x_0$
for $k = 0$, **to** $K - 1$ **do**
  Run Algorithm 2 for problem $f(x) + g(x)$, where $g(x)$ is a composite part.
  for $m = 0$, **to** $M - 1$ **do**
    Run Algorithm 2 up to desired accuracy for subproblem

$$\min_y \left(\Omega_p(f, \tilde{x}_k; y) + \frac{H_{p,f}}{p!} \|y - \tilde{x}_k\|^{p+1} + g(y) \right)$$

  end for
end for
return z_K

outer steps, it means that we need to compute $N_{out} = N_f$ derivatives of $f(x)$. Now we compute how much steps of inner method we need. Note that inner function has Lipshitz p-th derivative $H_{p,f} + L_g$. Also it is $(p+1)$-uniformly convex with σ_{p+1}. To compute σ_{p+1} we need to split $H_{p,f}$ into two parts $H_{p,f} = H_1 + H_2$, where the first part needs to make $\Omega_p(f, x; y) + \frac{H_1}{p!}\|y - x\|^{p+1}$ a convex function and the second part needs to make $\frac{H_2}{p!}\|y - x\|^{p+1}$ a uniformly convex term. Hence, from Lemma 1 we have $\sigma_{p+1} = \frac{H_2(p+1)2^{2-p}}{p!}$. We take $H_1 = H_2 = L_{p,f}$. As a result, the number of inner iterations equal to

$$\begin{aligned} N_{inn} &= \tilde{O}\left[\left(\frac{2L_{p,f} + L_{p,g}}{\frac{(p+1)L_{p,f}2^{2-p}}{p!}} \right)^{\frac{2}{3p+1}} \log\left(\frac{F(x_0) - F(x^*) + H_{p,f}R^{p+1}}{\varepsilon} \right) \right] \\ &= \tilde{O}\left[\left(\frac{2L_{p,f} + L_{p,g}}{\frac{(p+1)L_{p,f}2^{2-p}}{p!}} \right)^{\frac{2}{3p+1}} \right] \overset{L_{p,f} \le L_{p,g}}{=} \tilde{O}\left[\left(\frac{L_{p,g}}{L_{p,f}} \right)^{\frac{2}{3p+1}} \right] \end{aligned} \tag{11}$$

Hence the total number of inner iterations and total number of derivative's computations of $g(x)$ is

$$\begin{aligned} N_g = N_{out} \cdot N_{inn} &= \tilde{O}\left[\left(\frac{L_{p,f}R^{p+1-r}}{\sigma_r} \right)^{\frac{2}{3p+1}} \right] \cdot \tilde{O}\left[\left(\frac{L_{p,g}}{L_{p,f}} \right)^{\frac{2}{3p+1}} \right] \\ &= \tilde{O}\left[\left(\frac{L_{p,g}R^{p+1-r}}{\sigma_r} \right)^{\frac{2}{3p+1}} \right]. \end{aligned}$$

So we prove the theorem and split computation complexities. □

Note, that this framework also easily adapts to methods without accelerating like [26], [6]. But, unfortunately, it is much harder to adapt for other acceleration schemes. As we know, it is possible to adapt this framework for speed ups from [12] and [17] for $p \ge 2$, but for $p = 1$ it may arise some troubles because of adaptive inner regularisation and hence hard subproblem. As for [26] acceleration it also hard to adapt, because the inner subproblem is much harder with increasing complexity.

Also note that this framework can be generalized to the problem of the sum of m functions.

4 General convex functions

We consider (1) problem for convex functions.

If we will use Algorithm 1 for problem (1) we get next convergence speed. To reach $F(x_N) - F(x^*) \leq \varepsilon$, we need $N_f + N_g$ iterations, where

$$N_f = \tilde{O}\left[\left(\frac{L_{p,f}R^{p+1}}{\varepsilon}\right)^{\frac{2}{3p+1}}\right], \tag{12}$$

$$N_g = \tilde{O}\left[\left(\frac{L_{p,g}R^{p+1}}{\varepsilon}\right)^{\frac{2}{3p+1}}\right]. \tag{13}$$

Now we prove that the our framework split computation's complexities for convex functions.

Theorem 5. *Assume $f(x)$ and $g(x)$ are convex functions with Lipshitz p-th derivative ($p \geq 1$, $p+1 \geq q$) and $L_{p,f} < L_{p,g}$. Then by using our framework with $H_{p,f} = 2L_{p,f}$, method converges to $F(x_N) - F(x^*) \leq \varepsilon$ with N_f as* (12) *computations of derivatives $f(x)$ and N_g as* (13) *computation of derivatives $g(x)$.*

Proof. For the outer method 1 with constant $H_{p,f} = 2L_{p,f}$, we make

$$N_{out} = \tilde{O}\left[\left(\frac{2L_{p,f}R^{p+1}}{\varepsilon}\right)^{\frac{2}{3p+1}}\right]$$

outer steps, it means that we need to compute $N_{out} = N_f$ derivatives of $f(x)$. For inner method 1 to solve subproblem (4) similarly we has the same rate as (11) Hence the total number of inner iterations and total number of derivative's computations of $g(x)$ is

$$\begin{aligned} N_g = N_{out} \cdot N_{inn} &= \tilde{O}\left[\left(\frac{2L_{p,f}R^{p+1}}{\varepsilon}\right)^{\frac{2}{3p+1}}\right] \cdot \tilde{O}\left[\left(\frac{L_{p,g}}{L_{p,f}}\right)^{\frac{2}{3p+1}}\right] \\ &= \tilde{O}\left[\left(\frac{L_{p,g}R^{p+1}}{\varepsilon}\right)^{\frac{2}{3p+1}}\right]. \end{aligned}$$

So for convex function computation complexities are also splitting. □

5 Multi-Composite Tensor Method

The natural generalization of framework 3 is to use for the sum of two functions with different smoothness and hence different order of methods. But as we know, in the literature there is no method that works with the sum of two functions with different smoothness. We need to use tensor methods for the smallest order. To improve this situation we we introduce the new type of problem, where $f(x)$ and $g(x)$ have different smoothness order. Similar idea for the first and second order was in the paper [8]. Next we propose a tensor method to solve such problem with splitting the complexities.

We introduce a multi-composite tensor optimization problem.

$$F(x) = f(x) + g(x) + h(x),$$

where $h(x)$ is a simple proper closed convex function, $f(x)$ is a convex functions with Lipschitz q-th derivative and $g(x)$ is a convex functions with Lipschitz p-th derivative. By using Theorem 1 from [26] we can get for $f(x)$ if $H_{q,f} \geq qL_{q,f}$, that

$$\Omega_q(f, x; y) + \frac{H_{q,f}}{(q+1)!}\|y - x\|^{q+1}$$

is convex and

$$f(y) \leq \Omega_q(f, x; y) + \frac{H_{q,f}}{(q+1)!}\|y - x\|^{q+1}$$

Now we propose our method

$$T_{H_{q,f},H_{p,g}}(x) \in \operatorname*{Argmin}_y \left\{ \Omega_q(f, x; y) + \frac{H_{q,f}}{(q+1)!}\|y - x\|^{q+1} + \Omega_p(g, x; y) + \frac{H_{p,g}}{(p+1)!}\|y - x\|^{p+1} + h(y) \right\}$$

Then

$$x_{t+1} = T_{H_{q,f},H_{p,g}}(x_t) \tag{14}$$

One can see that our method based on method [26] and combine models of two functions. Next we start to prove, that our method converges and split the complexities.

We assume that exists at least one solution x_* of problem (1) and the level sets of F are bounded. By the first-order optimality condition for $T = T_{H_{q,f},H_{p,g}}(x)$ we get:

$$\nabla\Omega_q(f, x; T) + \frac{H_{q,f}(T - x)}{q!}\|T - x\|^{q-1} + \nabla\Omega_p(g, x; T) + \frac{H_{p,g}(T - x)}{p!}\|T - x\|^{p-1} + \partial h(T) = 0$$

For the proof we need next small lemma.

Lemma 2. *For any $x \in E$, $H_{q,f} \geq qL_{q,f}$ and $H_{p,g} \geq pL_{p,g}$, we have*

$$F(T_{H_{q,f},H_{p,g}}(x)) \leq \min_y \left\{ F(y) + \frac{H_{q,f} + L_{q,f}}{(q+1)!}\|y - x\|^{q+1} + \frac{H_{p,g} + L_{p,g}}{(p+1)!}\|y - x\|^{p+1} \right\} \tag{15}$$

Proof.

$$\begin{aligned} F(T_{H_{q,f},H_{p,g}}(x)) &\leq \min_y \left\{ \Omega_q(f, x; y) + \frac{H_{q,f}}{(q+1)!}\|y - x\|^{q+1} \right. \\ &\quad \left. + \Omega_p(g, x; y) + \frac{H_{p,g}}{(p+1)!}\|y - x\|^{p+1} + h(y) \right\} \\ &\overset{(3)}{\leq} \min_y \left\{ F(y) + \frac{H_{q,f} + L_{q,f}}{(q+1)!}\|y - x\|^{q+1} + \frac{H_{p,g} + L_{p,g}}{(p+1)!}\|y - x\|^{p+1} \right\} \end{aligned}$$

□

This leads us to the main theorem, that proves the convergence speed of our method.

Theorem 6. *If $f_q(x)$ is convex functions with Lipshitz constant $L_{q,f}$ for q-th derivative, $f_p(x)$ is convex functions with Lipshitz constant $L_{p,g}$ for p-th derivative; $H_{q,f} \geq qL_{q,f}$ and $H_{p,g} \geq pL_{p,g}$. α_t is chosen such that $\alpha_0 = 1$ and $\alpha_t \in [0;1]$ $t \geq 1$, then for any $t \geq 0$ for method* (14) *we have*

$$F(x_{t+1}) - F(x_*) \leq A_t \sum_{i=0}^{t} \left[C_f \frac{\alpha_i^{q+1}}{A_i} \|x_i - x_*\|^{q+1} + C_g \frac{\alpha_i^{p+1}}{A_i} \|x_i - x_*\|^{p+1} \right]$$

where

$$C_f = \frac{H_{q,f} + L_{q,f}}{(q+1)!}, \quad C_g = \frac{H_{p,g} + L_{p,g}}{(p+1)!};$$

$$A_t = \begin{cases} 1, & t = 0 \\ \prod\limits_{i=1}^{t} (1 - \alpha_i), & t \geq 1 \end{cases} \tag{16}$$

Proof. From (15)

$$\begin{aligned} F(x_{t+1}) &\leq \min_y \left\{ F(y) + \frac{H_{q,f} + L_{q,f}}{(q+1)!} \|y - x_t\|^{q+1} + \frac{H_{p,g} + L_{p,g}}{(p+1)!} \|y - x_t\|^{p+1} \right\} \\ &\leq F(y) + C_f \|y - x_t\|^{q+1} + C_g \|y - x_t\|^{p+1} \end{aligned}$$

If we take $y = x_t + \alpha_t(x_* - x_t)$, then by convexity

$$\begin{aligned} &F(x_{t+1}) \leq F(y) + C_f \alpha_t^{q+1} \|x_* - x_t\|^{q+1} + C_g \alpha_t^{p+1} \|x_* - x_t\|^{p+1} \\ &\leq (1 - \alpha_t) F(x_t) + \alpha_t F(x_*) + C_f \alpha_t^{q+1} \|x_* - x_t\|^{q+1} + C_g \alpha_t^{p+1} \|x_* - x_t\|^{p+1}. \end{aligned}$$

Hence

$$\begin{aligned} F(x_{t+1}) - F(x_*) \leq (1 - \alpha_t) &\left(F(x_t) - F(x_*)\right) \\ &+ C_f \alpha_t^{q+1} \|x_* - x_t\|^{q+1} + C_g \alpha_t^{p+1} \|x_* - x_t\|^{p+1} \end{aligned}$$

For $t = 0$ and $\alpha_0 = 1$ we get

$$F(x_1) - F(x_*) \leq C_f \|x_* - x_0\|^{q+1} + C_g \|x_* - x_0\|^{p+1}$$

For $t > 0$ we divide both sides by A_t:

$$\begin{aligned} \frac{1}{A_t}(F(x_{t+1}) - F(x_*)) &\leq \frac{(1 - \alpha_t)}{A_t} \left(F(x_t) - F(x_*)\right) \\ &+ C_f \frac{\alpha_t^{q+1}}{A_t} \|x_* - x_t\|^{q+1} + C_g \frac{\alpha_t^{p+1}}{A_t} \|x_* - x_t\|^{p+1} \\ &\overset{(16)}{\leq} \frac{1}{A_{t-1}} \left(F(x_t) - F(x_*)\right) \\ &+ C_f \frac{\alpha_t^{q+1}}{A_t} \|x_* - x_t\|^{q+1} + C_g \frac{\alpha_t^{p+1}}{A_t} \|x_* - x_t\|^{p+1} \end{aligned}$$

By summarising both sides we obtain (15) □

Next we can fix parameters of this theorem and get next corollary.

Corollary 7. *For method* (14) *and* $\alpha_t = \frac{p+1}{t+p+1}$ *we have*

$$F(x_{t+1}) - F(x_*) \leq E_q \frac{(H_{q,f} + L_{q,f})R^{q+1}}{(t+p+1)^q} + E_p \frac{(H_{p,g} + L_{p,g})R^{p+1}}{(t+p+1)^p} \tag{17}$$

where

$$E_k = \frac{(p+1)^{k+1}}{(k+1)!}, \quad k = \{q, p\}$$

Proof. We use

$$\begin{aligned} F(x_{t+1}) - F(x_*) &\leq A_t \sum_{i=0}^{t} \left[C_f \frac{\alpha_i^{q+1}}{A_i} \|x_i - x_*\|^{q+1} + C_g \frac{\alpha_i^{p+1}}{A_i} \|x_i - x_*\|^{p+1} \right] \\ &\overset{(6)}{\leq} C_f R^{q+1} \sum_{i=0}^{t} \frac{A_t \alpha_i^{q+1}}{A_i} + C_g R^{p+1} \sum_{i=0}^{t} \frac{A_t \alpha_i^{p+1}}{A_i} \end{aligned}$$

Now we compute these sums for $\alpha_t = \frac{p+1}{t+p+1}$:

$$\begin{aligned} A_t = \prod_{i=1}^{t}(1-\alpha_i) = \prod_{i=1}^{t} \frac{i}{i+p+1} = \frac{t!\,(p+1)!}{(t+p+1)!} = (p+1)! \prod_{i=1}^{p+1} \frac{1}{t+i} \\ \geq \frac{(p+1)!}{(t+1)^{p+1}} \end{aligned}$$

For the second sum we get

$$\begin{aligned} \sum_{i=1}^{t} \frac{A_t \alpha_i^{p+1}}{A_i} &= \sum_{i=1}^{t} \frac{(p+1)^{p+1} \prod_{j=1}^{p+1}(i+j)}{(i+p+1)^{p+1}(p+1)!} \cdot (p+1)! \prod_{i=1}^{p+1} \frac{1}{t+i} \\ &= (p+1)^{p+1} \sum_{i=1}^{t} \prod_{j=1}^{p+1} \frac{i+j}{i+p+1} \prod_{i=1}^{p+1} \frac{1}{t+i} \\ &\leq \frac{(p+1)^{p+1}}{(t+p+1)^p} \end{aligned}$$

For the first sum we get For second sum we have

$$\begin{aligned} \sum_{i=1}^{t} \frac{A_t \alpha_i^{q+1}}{A_i} &= \sum_{i=1}^{t} \frac{(p+1)^{q+1} \prod_{j=1}^{p+1}(i+j)}{(i+p+1)^{q+1}(p+1)!} \cdot (p+1)! \prod_{i=1}^{p+1} \frac{1}{t+i} \\ &= (p+1)^{q+1} \sum_{i=1}^{t} \frac{\prod_{j=1}^{p+1}(i+j)}{(i+p+1)^{q+1}} \cdot \prod_{i=1}^{p+1} \frac{1}{t+i} \\ &\leq \frac{(p+1)^{q+1}}{(t+p+1)^q}. \end{aligned}$$

From this two formulas for sums we get (17) □

Finally, we prove that our method converges with the desired speed and split the complexities. Note that this algorithm can be generalized for the sum of m functions.

6 Conclusion

In this paper, we consider the minimization of the sum of two functions $f + g$ each having Lipshitz p-th order derivatives with different Lipschitz constants. We propose a general framework to accelerate tensor methods by splitting computational complexities. As a result, we get near-optimal oracle complexity for each function in the sum separately for any $p \geq 1$, including the first-order methods. To be more precise, if the near optimal complexity to minimize f is $N_f(\varepsilon)$ iterations and to minimize g is $N_g(\varepsilon)$, then our method requires no more than $\tilde{O}(N_f(\varepsilon))$ oracle calls for f and $\tilde{O}(N_g(\varepsilon))$ oracle calls for g to minimze $f + g$. We prove, that our framework works with both convex and uniformly convex functions. To get this result, we additionally generalize near-optimal tensor methods for composite problems with inexact inner tensor step.

Further, we investigate the situation when the functions in the sum have Lipschitz derivatives of a different order. For this situation, we propose a generic way to separate the oracle complexity between the parts of the sum. It is the first tensor method that works with functions with different smoothness. Our method is not optimal, which leads to an open problem of the optimal combination of oracles of a different order.

References

[1] N. Agarwal and E. Hazan. Lower bounds for higher-order convex optimization. *arXiv preprint arXiv:1710.10329*, 2017.

[2] Y. Arjevani, O. Shamir, and R. Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, 178(1-2):327–360, 2019.

[3] A. Beznosikov, E. Gorbunov, and A. Gasnikov. Derivative-free method for decentralized distributed non-smooth optimization. *arXiv preprint arXiv:1911.10645*, 2019.

[4] S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford. Near-optimal method for highly smooth convex optimization. In *Conference on Learning Theory*, pages 492–507, 2019.

[5] P. Chebyshev. *collected works. Vol 5*. Strelbytskyy Multimedia Publishing, 2018.

[6] N. Doikov and Y. Nesterov. Local convergence of tensor methods. *arXiv preprint arXiv:1912.02516*, 2019.

[7] N. Doikov and Y. Nesterov. Minimizing uniformly convex functions by cubic regularization of newton method. *arXiv preprint arXiv:1905.02671*, 2019.

[8] N. Doikov and P. Richtárik. Randomized block cubic newton method. *arXiv preprint arXiv:1802.04084*, 2018.

[9] D. Dvinskikh and A. Gasnikov. Decentralized and parallelized primal and dual accelerated methods for stochastic convex programming problems. *arXiv preprint arXiv:1904.09015*, 2019.

[10] D. Dvinskikh, S. Omelchenko, A. Tiurin, and A. Gasnikov. Accelerated gradient sliding and variance reduction. *arXiv preprint arXiv:1912.11632*, 2019.

[11] P. Dvurechensky, A. Gasnikov, P. Ostroukhov, C. A. Uribe, and A. Ivanova. Near-optimal tensor methods for minimizing the gradient norm of convex function. *arXiv preprint arXiv:1912.03381*, 2019.

[12] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, and C. A. Uribe. Optimal tensor methods in smooth convex and uniformly convexoptimization. In *Conference on Learning Theory*, pages 1374–1391, 2019.

[13] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C. A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, et al. Near optimal methods for minimizing convex functions with lipschitz p-th derivatives. In *Conference on Learning Theory*, pages 1392–1393, 2019.

[14] E. Gorbunov, D. Dvinskikh, and A. Gasnikov. Optimal decentralized distributed algorithms for stochastic convex optimization. *arXiv preprint arXiv:1911.07363*, 2019.

[15] G. N. Grapiglia and Y. Nesterov. On inexact solution of auxiliary problems in tensor methods for convex optimization. *arXiv preprint arXiv:1907.13023*, 2019.

[16] G. N. Grapiglia and Y. Nesterov. Tensor methods for minimizing functions with h\"{o} lder continuous higher-order derivatives. *arXiv preprint arXiv:1904.12559*, 2019.

[17] B. Jiang, H. Wang, and S. Zhang. An optimal high-order tensor method for convex optimization. In *Conference on Learning Theory*, pages 1799–1801, 2019.

[18] L. V. Kantorovich. On newton's method. *Trudy Matematicheskogo Instituta imeni VA Steklova*, 28:104–144, 1949.

[19] G. Lan. Gradient sliding for composite optimization. *Mathematical Programming*, 159(1-2):201–235, 2016.

[20] G. Lan. Lectures on optimization methods for machine learning. *e-print*, 2019.

[21] G. Lan, S. Lee, and Y. Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, pages 1–48, 2018.

[22] G. Lan and Y. Ouyang. Accelerated gradient sliding for structured convex optimization. *arXiv preprint arXiv:1609.04905*, 2016.

[23] R. D. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[24] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[25] Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[26] Y. Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, pages 1–27, 2019.

[27] A. Rogozin and A. Gasnikov. Projected gradient method for decentralized optimization over time-varying networks. *arXiv preprint arXiv:1911.08527*, 2019.

[28] C. Song and Y. Ma. Towards unified acceleration of high-order algorithms under Hölder continuity and uniform convexity. *arXiv preprint arXiv:1906.00582*, 2019.

A Proof of Composite Accelerated Taylor Descent

This section is a rewriting of proof from [4], with adding composite part into the proof. Next theorem based on Theorem 2.1 from [4]

Theorem 8. *Let $(y_k)_{k\geq 1}$ be a sequence of points in $\mathbb{R}^d$ and $(\lambda_k)_{k\geq 1}$ a sequence in $\mathbb{R}_+$. Define $(a_k)_{k\geq 1}$ such that $\lambda_k A_k = a_k^2$ where $A_k = \sum_{i=1}^k a_i$. Define also for any $k \geq 0$, $x_k = x_0 - \sum_{i=1}^k a_i(\nabla f(y_i) + g'(y_i))$ and $\tilde{x}_k := \frac{a_{k+1}}{A_{k+1}}x_k + \frac{A_k}{A_{k+1}}y_k$. Finally assume if for some $\sigma \in [0,1]$*

$$\|y_{k+1} - (\tilde{x}_k - \lambda_{k+1}\nabla f(y_{k+1}))\| \leq \sigma \cdot \|y_{k+1} - \tilde{x}_k\|\,, \tag{18}$$

then one has for any $x \in \mathbb{R}^d$,

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

and

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

To prove this theorem we introduce auxiliaries lemmas based on lemmas 2.2-2.5 and 3.1 , lemmas 2.6 and 3.3 one can take directly from [4] without any changes.

Lemma 3. *Let $\psi_0(x) = \frac{1}{2}\|x - x_0\|^2$ and define by induction $\psi_k(x) = \psi_{k-1}(x) + a_k\Omega_1(F, y_k, x)$. Then $x_k = x_0 - \sum_{i=1}^k a_i(\nabla f(y_i) + g'(y_i))$ is the minimizer of ψ_k, and $\psi_k(x) \leq A_k F(x) + \frac{1}{2}\|x - x_0\|^2$ where $A_k = \sum_{i=1}^k a_i$.*

Lemma 4. *Let (z_k) be a sequence such that*

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

Then one has for any x,

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

Proof. One has (recall Lemma 3):

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

□

Lemma 5. *One has for any x,*

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

Proof. Firstly, by simple calculation we note that:

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

so that

$$\psi_{k+1}(x) - \psi_k(x_k) = a_{k+1}\Omega_1(F, y_{k+1}, x) + \frac{1}{2}\|x - x_k\|^2\,. \tag{19}$$

Now we want to make appear the term $A_{k+1}F(z_{k+1}) - A_kF(z_k)$ as a lower bound on the right hand side of (19) when evaluated at $x = x_{k+1}$. Using the inequality $\Omega_1(F, y_{k+1}, z_k) \leq f(z_k)$ we have:

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

which concludes the proof. □

Lemma 6. *Denoting* $\lambda_{k+1} := \frac{a_{k+1}^2}{A_{k+1}}$ *and* $\tilde{x}_k := \frac{a_{k+1}}{A_{k+1}}x_k + \frac{A_k}{A_{k+1}}y_k$ *one has:*

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

In particular, we have in light of (18)

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

Proof. We apply Lemma 5 with $z_k = y_k$ and $x = x_{k+1}$, and note that (with $\tilde{x} := \frac{a_{k+1}}{A_{k+1}}x + \frac{A_k}{A_{k+1}}y_k$):

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

This yields:

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

The value of the minimum is easy to compute. □

For the first conclusion in Theorem 8, it suffices to combine Lemma 6 with Lemma 4, and Lemma 2.5 from [4]. The second conclusion in Theorem 8 follows from Lemma 6 and Lemma 3.

The following lemma shows that minimizing the p^{th} order Taylor expansion (4) can be viewed as an implicit gradient step for some "large" step size:

Lemma 7. *Equation* (18) *holds true with* $\sigma = 1/2$ *for* (4)*, provided that one has:*

$$\frac{1}{2} \leq \lambda_{k+1} \frac{L_p \cdot \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1}\,. \tag{20}$$

Proof. Observe that the optimality condition gives:

$$\nabla_y f_p(y_{k+1}, \tilde{x}_k) + \frac{L_p \cdot (p+1)}{p!}(y_{k+1} - \tilde{x}_k)\|y_{k+1} - \tilde{x}_k\|^{p-1} + g'(y_{k+1}) = 0\,. \tag{21}$$

In particular we get:

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

By doing a Taylor expansion of the gradient function one obtains:

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

so that we find:

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

where we used (21) in the second last equation and we let $\eta := \lambda_{k+1}\frac{L_p \cdot \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!}$ in the last equation. The result follows from the assumption $1/2 \leq \eta \leq p/(p+1)$ in (20). □

Finally, if we replace $\|x^*\|$ by $\|x_0 - x^*\|$ in Lemma 3.3 and use Lemma 3.4 from [4] we prove Theorem 8.

B Inexact solution of the subproblem

Suppose that (4) can not be solved exactly. Assume that we can find only inexact solution $\tilde{y}_{k+1}$ satisfies

$$\left\|\nabla\left(f_p(\tilde{y}_{k+1}, \tilde{x}_k) + \frac{L_p}{p!}\|\tilde{y}_{k+1} - \tilde{x}_k\|^{p+1} + g(\tilde{y}_{k+1})\right)\right\| \leq \frac{L_p}{2p!}\|\tilde{y}_{k+1} - \tilde{x}_k\|^p. \tag{22}$$

In this case Lemma 7 should be corrected.

Lemma 8. *Equation* (18) *holds true with* $\sigma = 3/4$ *for* (22)*, provided that one has:*

$$\frac{1}{2} \leq \lambda_{k+1} \frac{L_p \cdot \|\tilde{y}_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1}\,.$$

Proof. Let's introduce

$$\Xi_{k+1} = \nabla \left(f_p(\tilde{y}_{k+1}, \tilde{x}_k) + \frac{L_p}{p!}\|\tilde{y}_{k+1} - \tilde{x}_k\|^{p+1} + g(\tilde{y}_{k+1}) \right).$$

The main difference with the proof of Lemma 7 is in the following line

$$\begin{aligned}
&\|\tilde{y}_{k+1} - (\tilde{x}_k - \lambda_{k+1}(\nabla f(\tilde{y}_{k+1}) + g'(\tilde{y}_{k+1})))\| \\
&\leq \lambda_{k+1}\frac{L_p}{p!}\|\tilde{y}_{k+1} - \tilde{x}_k\|^p + \\
&\left|\lambda_{k+1} - \frac{p!}{L_p \cdot (p+1) \cdot \|\tilde{y}_{k+1} - \tilde{x}_k\|^{p-1}}\right| \cdot \|\nabla_y f_p(\tilde{y}_{k+1}, \tilde{x}_k) + g'(\tilde{y}_{k+1})\| + \lambda_{k+1}\Xi_{k+1} \\
&\leq \|\tilde{y}_{k+1} - \tilde{x}_k\| \left(\lambda_{k+1}\frac{L_p}{p!}\|\tilde{y}_{k+1} - \tilde{x}_k\|^{p-1} + \left|\lambda_{k+1}\frac{L_p \cdot (p+1) \cdot \|\tilde{y}_{k+1} - \tilde{x}_k\|^{p-1}}{p!} - 1\right| \right) \\
&+ \|\tilde{y}_{k+1} - \tilde{x}_k\| \cdot \frac{1}{2p} \cdot \lambda_{k+1}\frac{L_p \cdot \|\tilde{y}_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!}.
\end{aligned}$$

To complete the proof it's left to notice that due to the (22)

$$\|\Xi_{k+1}\| \leq \frac{L_p}{2p!}\|\tilde{y}_{k+1} - \tilde{x}_k\|^p.$$

□

Based on (22) we try to relate the accuracy $\tilde{\varepsilon}$ we need to solve auxiliary problem to the desired accuracy ε for the problem (1). For this we use Lemma 2.1 from [15]. This Lemma guarantee that if

$$\left\| \nabla \left(f_p(\tilde{y}_{k+1}, \tilde{x}_k) + \frac{L_p}{p!}\|\tilde{y}_{k+1} - \tilde{x}_k\|^{p+1} + g(\tilde{y}_{k+1}) \right) \right\| \leq \frac{1}{4p(p+1)}\|\nabla F(\tilde{y}_{k+1})\|, \tag{23}$$

then (22) holds true. So it's sufficient to solve auxiliary problem in terms of (23).

Assume that $F(x)$ is r-uniformly convex function with constant σ_r ($r \geq 2$, $\sigma_r > 0$, see Definition 1), then from Lemma 2 [7] we have

$$F(\tilde{y}_{k+1}) - \min_{x \in E} F(x) \leq \frac{r-1}{r}\left(\frac{1}{\sigma_r}\right)^{\frac{1}{r-1}} \|\nabla F(\tilde{y}_{k+1})\|^{\frac{r}{r-1}}. \tag{24}$$

Inequalities (23), (24) give us guarantees that it's sufficient to solve auxiliary problem with the accuracy

$$\tilde{\varepsilon} = O\left(\left(\epsilon^{r-1}\sigma_r\right)^{\frac{1}{r}} \right)$$

in terms of criteria (23). Since auxiliary problem is every time r-uniformly convex we can apply (24) to auxiliary problem to estimate the accuracy in terms of function discrepancy. Anyway we will have that there is no need to think about it since the dependence of this accuracy are logarithmic. The only restrictive assumption we made is that $F(x)$ is r-uniformly convex. If this is not a case, like in Section 4, we may use regularisation tricks [11]. This lead us to $\sigma_2 \sim \varepsilon$. So the dependence $\tilde{\varepsilon}$ becomes worthier, but this doesn't change the main conclusion about possibility to skip the details concern the accuracy of the solution of auxiliary problem.

C CATD with restarts

The proof of the theorem 3.

Proof. As F is r-uniformly convex function we get

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

$$= \left(\frac{r c_p L_p R_k^{p+1}}{\sigma_r N_k^{\frac{3p+1}{2}}}\right)^{\frac{1}{r}} \overset{(7)}{\leq} \left(\frac{R_k^{p+1}}{2^r R_k^{p+1-r}}\right)^{\frac{1}{r}} = \frac{R_k}{2}.$$

Now we compute the total number of CATD steps.

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

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

□