Near-optimal tensor methods for minimizing the gradient norm of convex functions and accelerated primal-dual tensor methods

Pavel Dvurechensky[a], Petr Ostroukhov[b,c,d],Alexander Gasnikov[b,e,c], César A. Uribe[f], Anastasiya Ivanova[g,h]

[a] Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany; [b] Moscow Institute of Physics and Technology, Dolgoprudny, Russia; [c] Institute for Information Transmission Problems RAS, Moscow, Russia; [d] Mohamed bin Zayed University of Artificial Intelligence, Abu-Dhabi, UAE; [e] Skoltech, Moscow, Russia; [f] Department of Electrical and Computer Engineering, Rice University, Houston TX, USA; [g] HSE University, Moscow, Russian Federation; [h] Univ. Grenoble Alpes, LJK, 38000 Grenoble, France

ARTICLE HISTORY
Compiled August 11, 2023

ABSTRACT
Motivated, in particular, by the entropy-regularized optimal transport problem, we consider convex optimization problems with linear equality constraints, where the dual objective has Lipschitz p-th order derivatives, and develop two approaches for solving such problems. The first approach is based on the minimization of the norm of the gradient in the dual problem and then the reconstruction of an approximate primal solution. Recently, Grapiglia and Nesterov [22] showed lower complexity bounds for the problem of minimizing the gradient norm of the function with Lipschitz p-th order derivatives. Still, the question of optimal or near-optimal methods remained open as the algorithms presented in [22] achieve suboptimal bounds only. We close this gap by proposing two near-optimal (up to logarithmic factors) methods with complexity bounds $\tilde{O}(\varepsilon^{-2(p+1)/(3p+1)})$ and $\tilde{O}(\varepsilon^{-2/(3p+1)})$ with respect to the initial objective residual and the distance between the starting point and solution respectively. We then apply these results (having independent interest) to our primal-dual setting. As the second approach, we propose a direct accelerated primal-dual tensor method for convex problems with linear equality constraints, where the dual objective has Lipschitz p-th order derivatives. For this algorithm, we prove $\tilde{O}(\varepsilon^{-1/(p+1)})$ complexity in terms of the duality gap and the residual in the constraints. We illustrate the practical performance of the proposed algorithms in experiments on logistic regression, entropy-regularized optimal transport problem, and the minimal mutual information problem.

KEYWORDS
Tensor methods; gradient norm; nearly optimal methods; optimal transport; primal-dual methods

1. Introduction

The idea of using higher-order derivatives in optimization methods has been known since the 1970's [26], with increased interests recently [1, 4–6, 10, 46]. Using high-order oracles has been shown to have better oracle complexities provably. However,

CONTACT: Petr Ostroukhov: ostroukhov@phystech.edu, Pavel Dvurechensky: pavel.dvurechensky@wias-berlin.de

their main practical bottleneck was the requirement of solving an auxiliary problem at each iteration that involved minimizing a regularized Taylor expansion of the objective, which in general, is a non-convex problem. Nesterov in [37] showed that an appropriate regularization makes the auxiliary problem convex. Moreover, he proposed an efficient method for solving the corresponding subproblem for the third-order method. This motivated a resurgence of research that introduced high-order (also referred to as tensor) methods for convex [8, 9, 20, 21, 25, 27, 38, 39, 47] and non-convex settings [6, 10].

In this paper, we consider a convex optimization problem with linear equality constraints of the form

$$\min_{Ax=b} f(x), \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a convex function, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and its dual

$$\max_{\lambda\in\mathbb{R}^m} \left\{ \varphi(\lambda) := \langle \lambda, b\rangle + \max_{x\in Q} \left(-f(x) - \langle A^\intercal \lambda, x\rangle\right) \right\}, \tag{2}$$

where we assume that φ has Lipschitz-continuous p-th derivative with constant M_p.

One way to tackle this problem is by minimizing the gradient norm of the dual function [36]. In this case, the dual problem (2) is unconstrained, the gradient norm of the dual objective is equal to the primal constraints residual, and finding an ε-stationary point for the dual problem allows one to find an approximate solution to the primal problem, see the details in Section 2.

Recently in [22], the authors considered a more general setting of composite convex optimization problems with Hölder-continuous higher-order derivatives and proposed a set of methods for finding approximate stationary points. As a particular case of [22, Corollary 5.8], it follows that to find an ε-stationary point $\bar{x}$ such that $\|\nabla f(\bar{x})\|_2 \le \varepsilon$, their proposed method requires

$$O\big(\big(M_p R^p/\varepsilon\big)^{1/(p+1)}\big)$$

iterations with R being an estimate for the initial distance to the solution, i.e., $\|x_0 - x^*\|_2 \le R$. Moreover, as a particular case of [22, Corollary 5.10], it follows that to find an ε-stationary point, their proposed method needs

$$\tilde{O}\big(\big(M_p \Delta_0^p/\varepsilon^{p+1}\big)^{1/(p+1)}\big)$$

iterations with Δ_0 being an estimate for the initial objective residual, i.e., $f(x_0) - f^* \le \Delta_0$. However, *these complexity bounds do not match the corresponding lower bounds obtained in [22, Theorem 6.6] and [22, Theorem 6.8] respectively,* where the number of iterations required to find an ε-stationary point is of the order, respectively,

$$\Omega\big(\big(M_p R^p/\varepsilon\big)^{2/(3p+1)}\big) \qquad \text{and} \qquad \Omega\big(\big(M_p \Delta_0^p/\varepsilon^{p+1}\big)^{2/(3p+1)}\big).$$

In [7, 20, 21], the authors considered finding ε-approximate solution $\bar{x}$ in terms of the objective residual, i.e., such that $f(\bar{x}) - f^* \le \varepsilon$. They proposed a class of near-optimal methods up to a logarithmic factor for unconstrained minimization problem in the general convex setting and under the additional assumption of uniform convexity.

We build upon the algorithm, developed in [7], and propose methods for finding an approximate stationary point with near-optimal (up to a logarithmic multiplier) oracle complexities [22], see Table 1.

Table 1. Complexity of minimizing the gradient norm from [22] and from this paper (Ω means "lower bound").

Property	**Lower bound [22]**	**Upper Bound [22]**	**Upper Bound (this paper)**
$f(x_0) - f^* \leq \Delta_0$	$\Omega\Big(\frac{M_p \Delta_0^p}{\varepsilon^{p+1}}\Big)^{\frac{2}{3p+1}}$	$\tilde{O}\Big(\frac{M_p \Delta_0^p}{\varepsilon^{p+1}}\Big)^{\frac{1}{p+1}}$	$\tilde{O}\Big(\frac{M_p \Delta_0^p}{\varepsilon^{p+1}}\Big)^{\frac{2}{3p+1}}$
$\|x_0 - x^*\|_2 \leq R$	$\Omega\Big(\frac{M_p R^p}{\varepsilon}\Big)^{\frac{2}{3p+1}}$	$O\Big(\frac{M_p R^p}{\varepsilon}\Big)^{\frac{1}{p+1}}$	$\tilde{O}\Big(\frac{M_p R^p}{\varepsilon}\Big)^{\frac{2}{3p+1}}$

In addition, our methods can be used for strongly convex high-order smooth functions, and we provide complexity estimations for finding ε-approximate stationary point for this case. Moreover, we explain how our methods can be extended to obtain near-optimal methods for functions with Hölder-continuous high-order derivatives.

An alternative approach to tackle Problem (1), widely used in first-order methods [2, 3, 12, 17, 18, 23, 29–31, 41, 43–45, 48], constructs so-called primal-dual methods in which the main iterates are made for the dual problem with the goal being to find an ε-approximate solution to the dual problem. Then, the information generated while the dual algorithm works is used, e.g., by averaging, to reconstruct an ε-approximate solution to the primal problem. Motivated by such methods in the first-order setting, we propose an accelerated primal-dual high-order method that guarantees $O(1/k^{p+1})$ decay after k iterations both for the primal-dual gap and linear constraints infeasibility. To our knowledge, this is the first high-order primal-dual accelerated method. In particular, we are not aware of any primal-dual second-order methods.

This paper is organized as follows. We start in Section 2 with examples to motivate finding approximate stationary points of convex functions. We describe the entropy-regularized optimal transport problem and show that its structure provides a natural justification for tensor methods that exploit the high-order smoothness properties of the corresponding dual problem. Section 3 recalls some auxiliary results used later to prove our main results. Section 4.1 presents the near-optimal algorithm for finding approximate stationary points for the initial objective residual; near-optimal complexity bounds are shown explicitly. Section 4.2 shows the corresponding near-optimal algorithm for the initial variable residual; near-optimal complexity bounds are also shown. In Section 5, we propose and prove convergence rate guarantees of our accelerated primal-dual tensor method for problems with linear equality constraints. Section 6 shows some numerical results on the proposed algorithms for the logistic regression problem, entropy-regularized optimal transport problem, and minimization of "bad" functions, which give the lower bounds for the considered problem class. We provide numerical comparisons of the proposed tensor method for gradient norm minimization and primal-dual accelerated tensor method on entropy-regularized optimal transport and minimal mutual information problems. In addition, we compare numerically proposed algorithms with heuristical primal-dual modification of algorithm from [7]. Finally, conclusions and future work are presented in Section 7.

Notation: Let $p \geq 1$. We denote by $\nabla^p f(x)[h_1, ..., h_p]$ the directional derivative of function f at x along directions $h_i \in \mathbb{R}^n$, $i = 1, ..., p$. $\nabla^p f(x)[h_1, ..., h_p]$ is a symmetric p-linear form and its norm is defined as

$$\|\nabla^p f(x)\|_2 = \max_{h_1,...,h_p \in \mathbb{R}^n} \{\nabla^p f(x)[h_1, ..., h_p] : \|h_i\|_2 \leq 1, i = 1, ..., p\},$$

or equivalently

$$\|\nabla^p f(x)\|_2 = \max_{h \in \mathbb{R}^n} \{|\nabla^p f(x)[h, ..., h]| : \|h\|_2 \leq 1\}.$$

We denote $\|\cdot\|_2$ as the standard Euclidean norm, but our algorithm and derivations can be generalized for the Euclidean norm given by a general positive semi-definite matrix B. In what follows, we also use notation $\nabla^p f(x)[h]^p \equiv \nabla^p f(x)[h, ..., h]$. We consider convex, p times differentiable on $\mathbb{R}$ functions satisfying Lipschitz condition for p-th derivative

$$\|\nabla^p f(x) - \nabla^p f(y)\|_2 \leq M_p \|x - y\|_2, \qquad x, y \in \mathbb{R}^n. \tag{3}$$

Given a function f, numbers $p \geq 1$ and $M \geq 0$, define

$$\Phi_{x,p}(y) \triangleq \sum_{i=0}^{p} \frac{1}{i!} \nabla^i f(x)[y - x]^i,$$

$$\Omega_{x,p,M}(y) \triangleq \Phi_{x,p}(y) + \frac{M}{(p+1)!} \|y - x\|_2^{p+1}, \tag{4}$$

$$T^f_{p,M}(x) \in \operatorname{Arg\,min}_{y \in \mathbb{R}^n} \Omega_{x,p,M}(y). \tag{5}$$

The main reason for using such regularization in (4) is that, generally speaking, Taylor approximation of a convex function can be non-convex, and such regularization makes it convex [37] for sufficiently large M. Thus, from (3) and Taylor's theorem, it can be shown [6, Eqs. (2.6) and (2.7)], that

$$|f(y) - \Phi_{x,p}(y)| \leq \frac{M_p}{p!} \|y - x\|_2^{p+1}, \tag{6}$$

$$\|\nabla f(y) - \nabla \Phi_{x,p}(y)\|_2 \leq \frac{M_p}{(p-1)!} \|y - x\|_2^p. \tag{7}$$

2. A motivating example: problems with linear constraints

Let us consider a convex optimization problem with linear constraints

$$\min_{x \in Q \subseteq E} \{f(x) : Ax = b\}, \tag{8}$$

where E is a finite-dimensional real vector space, Q is a simple closed convex set, A is a given linear operator from E to some finite-dimensional real vector space H, $b \in H$ is given, f is a convex function on Q with respect to some chosen norm $\|\cdot\|_E$ on E.

The Lagrange dual problem for (8), written as a minimization problem, is

$$\min_{\lambda \in H^*} \left\{ \varphi(\lambda) := \langle \lambda, b \rangle + \max_{x \in Q} \left(-f(x) - \langle A^\intercal \lambda, x \rangle\right) \right\}. \tag{9}$$

We assume that the dual objective is smooth. In this case, by the Demyanov-Danskin

theorem [15], $\nabla\varphi(\lambda) = b - Ax(\lambda)$, where

$$x(\lambda) := \arg\max_{x \in Q} \left(-f(x) - \langle A^\intercal \lambda, x\rangle\right).$$

Having the dual formulation in (9) at hand, the following proposition justifies why minimizing the norm of a gradient is useful in the convex setting.

Proposition 2.1 (Lemma 1 in [19]). *Let $\lambda \in H^*$ be such that $-\langle \lambda, \nabla\varphi(\lambda)\rangle \leq \varepsilon_f$, and $\|\nabla\varphi(\lambda)\|_H \leq \varepsilon_{eq}$. Then*

$$f(x(\lambda)) - f^* \leq \varepsilon_f, \quad \|Ax(\lambda) - b\|_H \leq \varepsilon_{eq}. \tag{10}$$

Proposition 2.1 implies that if there is a method for the dual Problem (9) and this method generates a bounded sequence of iterates λ_k and a point λ_k s.t., the gradient of the dual objective is small, then, using the relation $x(\lambda_k)$ we can reconstruct an approximate solution to the primal problem. This is the general motivation for convex optimization methods for minimizing the objective gradient norm. Moreover, the complexity bound for the dual method directly translates to the complexity for solving the primal problem without any overhead.

To further motivate the high-order methods to minimize the objective gradient norm, we present a particular example of a smooth dual objective with Lipschitz continuous high-order derivatives. This example is the Entropy-regularized optimal transport problem [13, 14]. Next, we briefly describe the problem and the properties of the dual objective.

Consider two histograms $p, q \in \Sigma_n$ on a support of size n, where Σ_n is the standard simplex. Also, consider a matrix $M \in \mathbb{R}^{n\times n}_+$ which is symmetric and accounts for the "cost" of transportation such that M_{ij} is the cost of moving a unit of mass from bin i to bin j in the corresponding supports of distributions p and q. For example, given support points $(x_i)_{1\leq i\leq n}$ on the Euclidean space, one can consider $M_{ij} = \|x_i - x_j\|_2^2$, which corresponds to 2-Wasserstein distance.

The entropy-regularized optimal transport problem is defined as:

$$W_\gamma(p, q) \triangleq \min_{X \in U(p,q)} \{\langle M, X\rangle - \gamma E(X)\}, \tag{11}$$

where $\langle M, X\rangle$ is the Frobenius dot-product, $\gamma \geq 0$ is a regularization parameter, $E(X) \triangleq -\sum_{i,j} X_{ij} \ln(X_{ij})$, and U is the transportation polytope defined as

$$U(p, q) \triangleq \{X \in \mathbb{R}^{n\times n}_+ \mid X\mathbf{1}_n = p, X^\intercal \mathbf{1}_n = q\}.$$

It is known that Problem (11) is strongly convex and admits a unique optimal solution X^* [14]. If $\gamma = 0$ and $M_{ij} = \|x_i - x_j\|_2^r$, $W_\gamma(p, q)$ in (11) is known as the r-th power of the r-Wasserstein distance between p and q.

A standard way to deal with (11) is to write its dual as follows:

$$\begin{aligned}
&\min_{X\in U(p,q)} \langle M, X\rangle + \gamma\langle X, \ln X\rangle \\
&\quad = \min_{X\in\Sigma_{n^2}} \langle M, X\rangle + \gamma\langle X, \ln X\rangle + \max_{\xi,\eta} \{\langle \xi, p - X\mathbf{1}_n\rangle + \langle \eta, q - X^\intercal \mathbf{1}_n\rangle\} \\
&\quad = \max_{\xi,\eta} \left\{ \langle \xi, p\rangle + \langle \eta, q\rangle + \min_{X\in\Sigma_{n^2}} \{\langle M + \xi \mathbf{1}_n{}^\intercal + \mathbf{1}_n \mu^\intercal + \gamma \ln X, X\rangle\} \right\} \\
&\quad = \max_{\xi,\eta} -\gamma \ln \sum_{i,j=1}^{n} \exp\left(-\frac{1}{\gamma}(M_{ij} - \xi_i - \eta_j)\right) + \langle \xi, p\rangle + \langle \eta, q\rangle. \qquad (12)
\end{aligned}$$

In this case, the explicit dependence of the primal solution from the dual variables is given by

$$X(\xi,\eta) = \frac{\operatorname{diag}\left(\mathrm{e}^{\frac{\xi}{\gamma}}\right)\mathrm{e}^{-\frac{M}{\gamma}}\operatorname{diag}\left(\mathrm{e}^{\frac{\eta}{\gamma}}\right)}{\mathrm{e}^{\frac{\xi}{\gamma}}\mathrm{e}^{-\frac{M}{\gamma}}\mathrm{e}^{\frac{\eta}{\gamma}}}, \qquad (13)$$

where the function $\mathrm{e}(\cdot)$ indicates component-wise exponentiation of vectors and matrices, i.e., $[\mathrm{e}(A)]_{ij} = \exp(A_{ij})$. Also, for a vector a, $\operatorname{diag}(a)$ denotes a diagonal matrix with the vector a on the diagonal. We underline that as opposed to the standard dual problem derived in [13], we consider X to lie not in $\mathbb{R}_+^{n\times n}$, but rather in the standard simplex of the size n^2, the latter being the corollary of the marginal constraints $X\mathbf{1}_n = p$, $X^\intercal \mathbf{1}_n = q$ since $p, q \in \Sigma_n$. This allows us to obtain a high-order smooth dual objective with a softmax form, as we will show next. On the contrary, the dual problem in [13] has a sum of exponents in the dual objective, meaning that the derivatives are not Lipschitz-continuous.

To show the correspondence to a general primal and dual pair of Problems (8)–(9), let us assume without loss of generality that $E = \mathbb{R}^{n^2}$, $\|\cdot\|_E = \|\cdot\|_1$, and variable $x = \mathrm{vec}(X) \in \mathbb{R}^{n^2}$ to be the vector obtained from a matrix X by writing each column of X below the previous column. For the dual space we consider $H = \mathbb{R}^{2n}$, $\|\cdot\|_H = \|\cdot\|_2$. Also we set $f(x) = \langle M, X\rangle + \gamma\langle X, \ln X\rangle$, $Q = \Sigma_{n^2}$, $b^\intercal = (p^\intercal, q^\intercal)$, $A : \mathbb{R}^{n^2} \to \mathbb{R}^{2n}$ defined by the identity $(A\,\mathrm{vec}(X))^\intercal = ((X\mathbf{1}_n)^\intercal, (X^\intercal \mathbf{1}_n)^\intercal)$, and $\lambda^\intercal = (\xi^\intercal, \eta^\intercal)$. Note that the matrix A has the form

$$A = \begin{pmatrix} I_n & I_n & I_n & \dots \\ \mathbf{1}_n^\intercal & \mathbf{0}_n^\intercal & \mathbf{0}_n^\intercal & \dots \\ \mathbf{0}_n^\intercal & \mathbf{1}_n^\intercal & \mathbf{0}_n^\intercal & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix},$$

where I_n is the identity matrix, $\mathbf{0}_n^\intercal$ is the vector of all zeros. Using these notations, we can write the dual problem in (12) as

$$\begin{aligned}
&\max_{\lambda} -\gamma \ln \sum_{i,j=1}^{n} \exp\left(-\frac{[M - A^\intercal\lambda]_{ij}}{\gamma}\right) + \langle \lambda, b\rangle \\
&= \max_{\lambda} -\mathbf{smax}_\gamma(A^\intercal\lambda - M) + \langle \lambda, b\rangle \qquad (14) \\
&= \min_{\lambda} \mathbf{smax}_\gamma(A^\intercal\lambda - M) - \langle \lambda, b\rangle \qquad (15)
\end{aligned}$$

where

$$\mathbf{smax}_\gamma(y) \triangleq \gamma \log \left(\sum_{i=1}^{m} \exp(y_i/\gamma) \right). \tag{16}$$

More importantly, the following property holds.

Proposition 2.2 ([9, Theorem 3.4]). *Let $z \in \mathbb{R}^n$, $c \in \mathbb{R}^m$ and $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m$. Then the function $\boldsymbol{smax}_\gamma(\mathcal{A}z - c)$ is (order 3) $\frac{15}{\gamma^3}$-smooth with respect to $\|\cdot\|_{\mathcal{A}^\intercal \mathcal{A}}$.*

As a corollary, the dual objective in (15) has $(15/\gamma^3)$-Lipschitz-continuous third derivative w.r.t. $\|\cdot\|_{AA^\intercal}$. Equivalently to the first order Lipschitz constant from [35], we can write third order Lipschitz constant from Proposition 2.2 as $15\|A\|^4_{E\to H}/\gamma^3$. Due to our choice of norms in E and H, $\|A\|_{E\to H}$ is equal to the maximal Euclidean norm of the columns of matrix A. Thus, $\|A\|_{E\to H} = \sqrt{2}$.

We can conclude that minimizing the norm of the gradient of the dual objective allows one to obtain an approximate solution to the corresponding primal problem that estimates the optimal transport cost and optimal transportation plan in this case. Thus, having a fast method that exploits the high-order smoothness of the dual problem can provide efficient algorithms for the computation of Sinkhorn distance [13] defined as the solution to entropy regularized optimal transport problem.

3. Preliminaries

In this section, we present a series of auxiliary results that will later enable the development of our near-optimal algorithms, which will be presented in Section 4. The reader might skip this section and revisit it for proof details.

We measure the complexity of algorithms in the number of calls to the oracle of the objective function. By oracle of some objective f we mean some mapping $x \mapsto \{f(x), \nabla f(x), ..., \nabla^p f(x)\}, \forall x \in \mathrm{dom} f,\ p \geq 1$.

To make the paper self-contained, in this section, we recall the near-optimal tensor methods for minimization of convex objective functions with Lipschitz-continuous p-th derivative [7].

Theorem 3.1 (Theorem 1 in [7]). *Let f be a convex function with M_p-Lipschitz p-th derivative. Assume, that exists $R > 0 : \|x_0 - x^*\|_2 \leq R$, and let $c_p = 2^{p-1}(p+1)^{\frac{3p+1}{2}}/(p-1)!$. Then, for all $N \geq 0$, the output of Algorithm 1 has the following property*

$$f(y_N) - f(x^*) \leq \frac{c_p M_p R^{p+1}}{N^{\frac{3p+1}{2}}}. \tag{18}$$

Moreover, each iteration k requires $O\left(\ln(1/\varepsilon)\right)$ oracle calls.

At the core of the result in Theorem 3.1, the authors in [7] use the following auxiliary result that we will later use in our proofs. We restate this result for completeness.

Lemma 3.2 (Lemma 11 from [7]). *Let $c_p = 2^{p-1}(p+1)^{\frac{3p+1}{2}}/(p-1)!$, and $k \geq 0$.*

Algorithm 1 Accelerated Taylor Descent [7, Algorithm 1]

Require: N — iteration number.

1: Set $A_0 = 0$, $x_0 = y_0 = 0$

2: **for** $k = 0, 1, 2, \ldots, N-1$ **do**

3: Compute $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ such that

$$\frac{1}{2} \leq \lambda_{k+1} \frac{M_p \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1}, \tag{17}$$

where

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}, \quad A_{k+1} = A_k + a_{k+1},$$
$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k, \quad y_{k+1} = T^f_{p, pM_p}(\tilde{x}_k).$$

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

5: **end for**

6: **return** y_N

Then, A_k from Algorithm 1 has the following property

$$A_k \geq \frac{1}{c_p M_p R^{p-1}} k^{\frac{3p+1}{2}}. \tag{19}$$

The following statement from [32] also holds for Algorithm 1.

Theorem 3.3 (Theorem 3.6 in [32])**.** *Let sequence $(x_k, \tilde{x}_k, y_k), k \geq 0$ be generated by Algorithm 1, and define $R := \|x_0 - x^*\|_2$. Then for all $N \geq 0$*

$$\frac{1}{2}\|x_N - x^*\|_2^2 + A_N \left(f(y_N) - f(x^*)\right) + \frac{1}{4} \sum_{k=1}^{N} A_k L_{k-1} \|y_k - \tilde{x}_{k-1}\|_2^2 \leq \frac{R^2}{2}, \tag{20}$$

$$f(y_N) - f(x^*) \leq \frac{R^2}{2A_N}, \quad \|x_N - x^*\|_2 \leq R, \tag{21}$$

$$\sum_{k=1}^{N} A_k L_{k-1} \|y_k - \tilde{x}_{k-1}\|_2^2 \leq 2R^2. \tag{22}$$

Algorithm 1 requires intermediate steps to find L_k and y_{k+1}. Since they depend on each other, we need to find them iteratively with a binary search procedure described in [7, Section 4].

Since we know that $\Omega_{x,p,M}(y)$ is convex, then $\tilde{z} = T^f_{p,M}(x)$ exists. Thus,

$$\forall x \in \mathbb{R}^n \Rightarrow \Omega_{x,p,M}(\tilde{z}) \overset{(5)}{=} \min_{y \in \mathbb{R}^n} \Omega_{x,p,M}(y) \leq \Omega_{x,p,M}(x) \overset{(4)}{=} f(x). \tag{23}$$

We use this inequality to prove the following lemma, which is a particular case of [22,

Lemma 5.2] with $\nu = 1$, $\theta = 0$, and $\varphi = 0$.

Lemma 3.4 (Lemma 5.2 in [22]). *Let $p \geq 1$, $M_p < \infty$, $M \geq (p+2)M_p$ and let for some $x \in \mathbb{R}^n$*

$$\tilde{z} = T^f_{p,M}(x).$$

Then,

$$f(x) - f(\tilde{z}) \geq \frac{1}{4(p+2)!M^{\frac{1}{p}}} \|\nabla f(\tilde{z})\|_2^{\frac{p+1}{p}}.$$

Proof. From triangle inequality, (7) and definition of $\tilde{z}$, we get

$$\begin{aligned}
\|\nabla f(\tilde{z})\|_2 &= \|\nabla f(\tilde{z}) - \nabla \Phi_{x,p}(\tilde{z}) + \nabla \Phi_{x,p}(\tilde{z}) - \nabla \Omega_{x,p,M}(\tilde{z}) + \nabla \Omega_{x,p,M}(\tilde{z})\|_2 \\
&\leq \|\nabla f(\tilde{z}) - \nabla \Phi_{x,p}(\tilde{z})\|_2 + \|\nabla \Phi_{x,p}(\tilde{z}) - \nabla \Omega_{x,p,M}(\tilde{z})\| + \|\nabla \Omega_{x,p,M}(\tilde{z})\|_2 \\
&\overset{(7)}{\leq} \frac{M_p}{(p-1)!}\|\tilde{z} - x\|_2^p + \frac{M}{p!}\|\tilde{z} - x\|_2^p = \left(\frac{pM_p}{p!} + \frac{M}{p!}\right)\|\tilde{z} - x\|_2^p \\
&\leq 2M\|\tilde{z} - x\|_2^p. \qquad (24)
\end{aligned}$$

Next, from (6), (23) follows

$$\begin{aligned}
f(\tilde{z}) &\overset{(6)}{\leq} \Phi_{x,p}(\tilde{z}) + \frac{M_p}{p!}\|\tilde{z} - x\|_2^{p+1} = \Phi_{x,p}(\tilde{z}) + \frac{(p+1)M_p}{(p+1)!}\|\tilde{z} - x\|_2^{p+1} \\
&= \Phi_{x,p}(\tilde{z}) + \frac{M}{(p+1)!}\|\tilde{z} - x\|_2^{p+1} - \frac{(M - (p+1)M_p)}{(p+1)!}\|\tilde{z} - x\|_2^{p+1} \\
&= \Omega_{x,p,M}(\tilde{z}) - \frac{(M - (p+1)M_p)}{(p+1)!}\|\tilde{z} - x\|_2^{p+1} \\
&\overset{(23)}{\leq} f(x) - \frac{(M - (p+1)M_p)}{(p+1)!}\|\tilde{z} - x\|_2^{p+1}.
\end{aligned}$$

Since $M \geq (p+2)M_p \Leftrightarrow \frac{1}{p+2}M \geq M_p$, we get

$$\begin{aligned}
f(x) - f(\tilde{z}) &\geq \frac{(M - (p+1)M_p)}{(p+1)!}\|\tilde{z} - x\|_2^{p+1} \\
&\geq \frac{(M - \frac{p+1}{p+2}M)}{(p+1)!}\|\tilde{z} - x\|_2^{p+1} \\
&= \frac{M}{(p+2)!}\|\tilde{z} - x\|_2^{p+1}. \qquad (25)
\end{aligned}$$

If we combine (24) and (25), we obtain the final result for all $p \geq 1$

$$f(x) - f(\tilde{z}) \geq \frac{M}{(p+2)!}\|\tilde{z} - x\|_2^{p+1} = \frac{M}{(p+2)!}(\|\tilde{z} - x\|_2^p)^{(p+1)/p}$$
$$\overset{(24)}{\geq} \frac{M}{(p+2)!}\left(\frac{\|\nabla f(\tilde{z})\|_2}{2M}\right)^{\frac{p+1}{p}} = \frac{\|\nabla f(\tilde{z})\|_2^{\frac{p+1}{p}}}{2^{\frac{p+1}{p}} M^{\frac{1}{p}}(p+2)!}$$
$$\geq \frac{\|\nabla f(\tilde{z})\|_2^{\frac{p+1}{p}}}{4M^{\frac{1}{p}}(p+2)!}.$$

□

4. Near-optimal tensor methods for gradient norm minimization

In this section, we will build upon Algorithm 1 to develop near-optimal tensor methods for gradient norm minimization of convex functions. This section is divided into two parts: first, we develop near-optimal tensor methods with respect to an estimate of the initial objective residual in Subsection 4.1 presented in Algorithm 2, then in Subsection 4.2, we develop near-optimal tensor methods with respect to an estimate of the initial argument residual presented in Algorithm 3. Note that both proposed algorithms have Algorithm 1 at their core, and rely on the bounds presented in Section 3.

4.1. Near-optimal tensor methods with respect to the initial objective residual

In this subsection, we build up from Algorithm 1 to develop a near-optimal algorithm for which we can provide explicit complexity bounds for approximating a stationary point. The obtained oracle complexity bound matches up to a logarithmic factor the lower complexity bound presented in [22]. This subsection focuses on complexity bounds that depend on the initial objective residual. Thus, the basic assumption is that the starting point x_0 satisfies $f(x_0) - f^* \leq \Delta_0$.

Theorem 4.1. *Let $p \geq 2$. Assume the function f is convex, p times differentiable on $\mathbb{R}^n$ with M_p-Lipschitz p-th derivative. Assume, that $\Delta_0 > 0$ is such that $f(x_0) - f^* \leq \Delta_0$. Let $\tilde{z}$ be generated by Algorithm 2. Then*

$$\|\nabla f(\tilde{z})\|_2 \leq \varepsilon,$$

and the total number of iterations of Algorithm 1 required by Algorithm 2 is

$$O\left(\frac{M_p^{\frac{2}{3p+1}}}{\varepsilon^{\frac{2(p+1)}{3p+1}}}\Delta_0^{\frac{2p}{3p+1}} + \log_2 \frac{2^{\frac{4p-3}{p+1}}\Delta_0(pM_p)^{\frac{1}{p}}(p+1)!}{\varepsilon^{\frac{p}{p+1}}}\right).$$

Moreover, the total oracle complexity is within a $O\left(\ln \frac{1}{\varepsilon}\right)$ factor of the above iteration complexity due to the use of binary search in Algorithm 1.

Algorithm 2 Near-optimal algorithm with respect to initial objective residual

Require: $p \geq 2$, M_p, x_0, $\Delta_0 : f(x_0) - f^* \leq \Delta_0$, $\varepsilon > 0$.

1: **Define**:

$$k = 0, \quad M_\mu = (p+2)M_p, \quad \mu = \frac{\varepsilon^2}{32\Delta_0}, \quad \tilde{\varepsilon} = \frac{(\varepsilon/2)^{\frac{p+1}{p}}}{4(p+2)!M_\mu^{\frac{1}{p}}},$$
$$z_0 = x_0, \quad f_\mu(x) = f(x) + \frac{\mu}{2}\|x - x_0\|_2^2.$$

2: **while** $\Delta_k \geq \tilde{\varepsilon}$ where $\Delta_k = \Delta_0 \cdot 2^{-k}$. **do**

3: Set z_{k+1} as the output of Algorithm 1 applied to $f_\mu(x)$ starting from z_k and run for N_k steps, where N_k is such that $A_{N_k} \geq 2/\mu$.

4: $k = k + 1$.

5: **end while**

6: **Find** $\tilde{z} = T^{f_\mu}_{p,\, M_\mu}(z_k)$.

7: **return** $\tilde{z}$.

Proof. By definition of $f_\mu(x)$:

$$f_\mu(x_0) - f_\mu(x_\mu^*) = f(x_0) - f(x_\mu^*) - \frac{\mu}{2}\|x_\mu^* - x_0\|_2^2 \leq f(x_0) - f(x^*) \leq \Delta_0,$$

Where x_μ^* is the minimum of $f_\mu(x)$. So, for $k = 0$ we have $f_\mu(z_k) - f_\mu(x_\mu^*) \leq \Delta_k$. Let us assume that $f_\mu(z_k) - f_\mu(x_\mu^*) \leq \Delta_k$ and show that $f_\mu(z_{k+1}) - f_\mu(x_\mu^*) \leq \Delta_{k+1}$. As you can see, we use Algorithm 1 inside Algorithm 2 and restart it every time $A_{N_k} \geq 2/\mu$. We can do this due to the strong convexity of f_μ. From (21), strong convexity and this restart condition it holds that

$$\begin{aligned} f_\mu(z_{k+1}) - f_\mu(x_\mu^*) &\overset{(21)}{\leq} \frac{\|z_k - x_\mu^*\|_2^2}{2A_{N_k}} \leq \frac{1}{2A_{N_k}} \left(\frac{2(f_\mu(z_k) - f_\mu(x_\mu^*))}{\mu} \right) \\ &\leq \frac{\Delta_k}{\mu A_{N_k}} \leq \frac{\Delta_k}{2} = \Delta_{k+1}. \end{aligned} \tag{26}$$

Thus, $f_\mu(z_k) - f_\mu(x_\mu^*) \leq \Delta_k$ for all $k \geq 0$.

Although such a stopping criterion is useful for numerical experiments, it is not obvious how to derive a theoretical upper bound for the number of steps of Algorithm 1. We can use (19) and (26) to obtain an upper bound for the number $\tilde{N}_k$ of steps of Algorithm 1 sufficient to fulfill this stopping criterion. Denote by $A_{\tilde{N}_k}$ such constant A_N, which we get after $\tilde{N}_k$ steps of the Algorithm 1. Then

$$f_\mu(z_{k+1}) - f_\mu(x_\mu^*) \overset{(26)}{\leq} \frac{\Delta_k}{\mu A_{\tilde{N}_k}}.$$

From strong convexity, we know that

$$\|z_k - x_\mu^*\|_2 \leq \sqrt{\frac{2}{\mu}\Delta_k}.$$

Thus, we can choose $R_k = \sqrt{(2/\mu)\Delta_k}$.

From (19) we can choose N_k to fulfill the stopping criterion $A_{\tilde{N}_k} \geq 2/\mu$:

$$\tilde{N}_k = \max\left\{\left\lceil\left(\frac{2c_pM_p2^{\frac{p+1}{2}}}{\mu^{\frac{p+1}{2}}}\Delta_k^{\frac{p-1}{2}}\right)^{\frac{2}{3p+1}}\right\rceil, 1\right\}. \tag{27}$$

Therefore, we get

$$f_\mu(z_{k+1}) - f_\mu(x_\mu^*) \overset{(26)}{\leq} \frac{\Delta_k}{\mu A_{\tilde{N}_k}} \leq \frac{c_pM_p}{\tilde{N}_k^{\frac{3p+1}{2}}}\left(\frac{2\Delta_k}{\mu}\right)^{\frac{p+1}{2}} \leq \frac{\Delta_k}{2}.$$

Next, we estimate $\|\nabla f_\mu(\tilde{z})\|_2$. Since $\tilde{z} = T_{p,M_\mu}^{f_\mu}(z_k)$, and according to Lemma 3.4, we have

$$f_\mu(z_k) - f_\mu(\tilde{z}) \geq \frac{1}{4(p+2)!M_\mu^{\frac{1}{p}}}\|\nabla f_\mu(\tilde{z})\|_2^{\frac{p+1}{p}}. \tag{28}$$

At the same time, by the stopping criterion in Algorithm 2,

$$f_\mu(z_k) - f_\mu(\tilde{z}) \leq f_\mu(z_k) - f_\mu(x_\mu^*) \leq \Delta_k \leq \tilde{\varepsilon}. \tag{29}$$

By the definition of $\tilde{\varepsilon}$ and (28), (29), we have that

$$\|\nabla f_\mu(\tilde{z})\|_2 \leq \frac{\varepsilon}{2}. \tag{30}$$

Since the right-hand side of (28) is non-negative, we can state that

$$f_\mu(\tilde{z}) \leq f_\mu(z_k). \tag{31}$$

By definition, f_μ is μ-strongly convex and, using (31), we get

$$\frac{\mu}{2}\|x_\mu^* - x_0\|_2^2 \leq f_\mu(x_0) - f_\mu(x_\mu^*) \leq \Delta_0, \tag{32}$$

$$\frac{\mu}{2}\|\tilde{z} - x_\mu^*\|_2^2 \leq f_\mu(\tilde{z}) - f_\mu(x_\mu^*) \overset{(31)}{\leq} f_\mu(z_k) - f_\mu(x_\mu^*) \leq \Delta_k \leq \Delta_0. \tag{33}$$

Applying triangle inequality to the sum of (32) and (33), we get

$$\frac{\mu}{2}\|\tilde{z} - x_0\|_2^2 \leq \mu\big(\|x_\mu^* - x_0\|_2^2 + \|\tilde{z} - x_\mu^*\|_2^2\big) \leq 4\Delta_0,$$

and

$$\|\tilde{z} - x_0\|_2 \leq 2\sqrt{\frac{2\Delta_0}{\mu}}.$$

By definition of μ in Algorithm 2, we have

$$\mu\|\tilde{z} - x_0\|_2 \leq \mu \cdot 2\sqrt{\frac{2\Delta_0}{\mu}} = 2\sqrt{2\mu\Delta_0} = \frac{\varepsilon}{2}. \tag{34}$$

Finally, according to the definition of f_μ, (30), (34) and triangle inequality, we get

$$\|\nabla f(\tilde{z})\|_2 \leq \|\nabla f_\mu(\tilde{z})\|_2 + \mu\|\tilde{z} - x_0\|_2 \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

It remains to upper bound the total number of steps of Algorithm 1. Denote $\tilde{c} = \left(2c_p 2^{\frac{p+1}{2}}\right)^{\frac{2}{3p+1}}$ and aggregate $\tilde{N}_k$'s from (27)

$$\sum_{i=0}^{k} \tilde{N}_i \leq \tilde{c}\frac{M_p^{\frac{2}{3p+1}}}{\mu^{\frac{p+1}{3p+1}}} \sum_{i=0}^{k} (\Delta_0 \cdot 2^{-i})^{\frac{p-1}{3p+1}} + k \leq \tilde{c}\frac{M_p^{\frac{2}{3p+1}}}{\mu^{\frac{p+1}{3p+1}}} \Delta_0^{\frac{p-1}{3p+1}} \cdot \sum_{i=0}^{k} 2^{-i\frac{p-1}{3p+1}} + k$$

Since $p \geq 2$, $\sum_{i=0}^{k} 2^{-i\frac{p-1}{3p+1}}$ is a geometric progression with a common ratio of less than one. Therefore, we can upper bound its partial sum by its infinite sum:

$$\sum_{i=0}^{k} 2^{-i\frac{p-1}{3p+1}} \leq \frac{2}{1 - 2^{-\frac{p-1}{3p+1}}} \leq 2 \cdot 16 = 32. \tag{35}$$

Thus, we get

$$\begin{aligned}\sum_{i=0}^{k} \tilde{N}_i &\leq \tilde{c}\frac{M_p^{\frac{2}{3p+1}}}{\mu^{\frac{p+1}{3p+1}}} \Delta_0^{\frac{p-1}{3p+1}} \cdot \sum_{i=0}^{k} 2^{-i\frac{p-1}{3p+1}} + k \leq 32\tilde{c}\frac{M_p^{\frac{2}{3p+1}}}{\mu^{\frac{p+1}{3p+1}}} \Delta_0^{\frac{p-1}{3p+1}} + \log_2 \frac{\Delta_0}{\tilde{\varepsilon}} \\ &= O\left(\frac{M_p^{\frac{2}{3p+1}}}{\varepsilon^{\frac{2(p+1)}{3p+1}}} \Delta_0^{\frac{2p}{3p+1}} + \log_2 \frac{2^{\frac{4p-3}{p+1}} \Delta_0 (pM_p)^{\frac{1}{p}} (p+1)!}{\varepsilon^{\frac{p}{p+1}}}\right).\end{aligned} \tag{36}$$

According to Theorem 3.1, the total number of oracle calls is within the $O\left(\ln(1/\varepsilon)\right)$ factor from the number of iterations of Algorithm 1. This completes the proof. □

If we omit the dominated factors in the result (36), we obtain the complexity bound

$$\tilde{O}\left(\frac{M_p \Delta_0^p}{\varepsilon^{p+1}}\right)^{\frac{2}{3p+1}},$$

where the $\tilde{O}$ notation hides an additional multiplicative logarithmic factor. We can conclude that this bound coincides with the lower bound

$$\Omega\Big(\frac{M_p \Delta_0^p}{\varepsilon^{p+1}}\Big)^{\frac{2}{3p+1}},$$

from [22] up to logarithmic and constant factors.

4.2. Near-optimal tensor methods with respect to the initial variable residual

In this subsection, we build up from Algorithm 1 to develop a near-optimal algorithm, we provide explicit complexity bounds for approximating a stationary point. The obtained oracle complexity bound matches the lower bound presented in [22] up to a logarithmic factor. The basic assumption is that the starting point x_0 satisfies $\|x_0 - x^*\|_2 \leq R$.

Algorithm 3 Near-optimal algorithm for initial argument residual

Require: $p \geq 2$, M_p, x_0, $R : \|x_0 - x^*\|_2 \leq R$, $\varepsilon > 0$.

1: **Define**:

$$k = 0, \quad M_\mu = (p+2)M_p, \quad \mu = \frac{\varepsilon}{4R}, \quad \tilde{\varepsilon} = \frac{(\varepsilon/2)^{\frac{p+1}{p}}}{4(p+2)!M_\mu^{\frac{1}{p}}},$$
$$z_0 = x_0, \quad f_\mu(x) = f(x) + \frac{\mu}{2}\|x - x_0\|_2^2.$$

2: **while** $\mu R_k^2/2 \geq \tilde{\varepsilon}$, where $R_k = R \cdot 2^{-k}$ **do**

3: Set $z_{k+1} = y_{N_k}$ as the output of Algorithm 1 applied to $f_\mu(x)$ starting from z_k and run for N_k steps, where N_k is such that $A_{N_k} \geq 4/\mu$.

4: $k = k + 1$.

5: **end while**

6: **Find** $\tilde{z} = T^{f_\mu}_{p,\, M_\mu}(z_k)$.

7: **return** $\tilde{z}$.

Theorem 4.2. *Let $p \geq 2$. Assume the function f is convex, p times differentiable on $\mathbb{R}^n$ with M_p-Lipschitz p-th derivative. Assume that there exists $R > 0$ is such that $\|x_0 - x^*\|_2 \leq R$. Let $\tilde{z}$ be generated by Algorithm 3. Then*

$$\|\nabla f(\tilde{z})\|_2 \leq \varepsilon \tag{37}$$

and the total number of iterations of Algorithm 1 required by Algorithm 3 is

$$O\left(\frac{M_p^{\frac{2}{3p+1}} R^{\frac{2p}{3p+1}}}{\varepsilon^{\frac{2}{3p+1}}} + \log \frac{2^{\frac{p}{p+1}}(p+1)!(pM_p)^{\frac{1}{p}}}{\varepsilon^{\frac{1}{p+1}}}\right).$$

Moreover, the total oracle complexity is within a $O(\ln(1/\varepsilon))$ factor of the above iteration complexity.

Proof. By definition of $f_\mu(x)$, we have

$$f(x_\mu^*) + \frac{\mu}{2}\|x_\mu^* - x_0\|_2^2 = f_\mu(x_\mu^*) \leq f_\mu(x^*) = f(x^*) + \frac{\mu}{2}\|x^* - x_0\|_2^2 \leq f(x_\mu^*) + \frac{\mu}{2}\|x^* - x_0\|_2^2. \tag{38}$$

Hence, $\|x_\mu^* - x_0\|_2^2 \leq \|x^* - x_0\|_2^2 \leq R^2$. So, for $k = 0$ we have $\|x_\mu^* - z_k\|_2 \leq R_k$.

Let us assume that $\|x_\mu^* - z_k\|_2 \leq R_k$ and show that $\|x_\mu^* - z_{k+1}\|_2 \leq R_{k+1}$. Again, just like in Algorithm 2, we use Algorithm 1 inside Algorithm 3 and restart it every time $A_{N_k} \geq \frac{4}{\mu}$. We can do this due to the strong convexity of f_μ. From strong convexity,

(21) and this restart condition, it holds that

$$\frac{\mu}{2}\|z_{k+1} - x_\mu^*\|_2^2 \leq f_\mu(z_{k+1}) - f_\mu(x_\mu^*) \overset{(21)}{\leq} \frac{\|z_k - x_\mu^*\|_2^2}{2A_{N_k}} \leq \frac{R_k^2}{2A_{N_k}} \leq \frac{\mu R_k^2}{8} = \frac{\mu R_{k+1}^2}{2}. \quad (39)$$

Thus, $\|z_{k+1} - x_\mu^*\|_2 \leq R_{k+1},\ f_\mu(z_k) - f_\mu(x_\mu^*) \leq \frac{\mu R_k^2}{2}$ for all $k \geq 0$.

Since $f_\mu(x)$ is strongly convex and $\forall k \geq 0 \Rightarrow \|z_k - x_\mu^*\|_2 \leq R_k$, we can apply restarts technique. In the same way, as in the previous subsection, we can theoretically estimate the upper bound $\tilde{N}_k$ on the number of iterations for Algorithm 1 before the stopping criterion $A_{\tilde{N}_k} \geq 4/\mu$ holds:

$$f_\mu(z_{k+1}) - f_\mu(x_\mu^*) \overset{(39)}{\leq} \frac{R_k^2}{2A_{\tilde{N}_k}}.$$

Therefore, if we choose

$$\tilde{N}_k = \max\left\{\left\lceil\left(\frac{8c_p M_p R_k^{p-1}}{\mu}\right)^{\frac{2}{3p+1}}\right\rceil, 1\right\}, \quad (40)$$

then from (19) we see that this number of steps is sufficient to fulfill the stopping criterion $A_{\tilde{N}_k} \geq 4/\mu$:

$$f_\mu(z_{k+1}) - f_\mu(x_\mu^*) \leq \frac{R_k^2}{2A_{\tilde{N}_k}} \leq \frac{cM_p R_k^{p+1}}{\tilde{N}_k^{\frac{3p+1}{2}}} \leq \frac{\mu R_{k+1}^2}{2}.$$

Next, we estimate $\|\nabla f_\mu(\tilde{z})\|_2$. Since $\tilde{z} = T_{p,M_\mu}^{f_\mu}(z_k)$, and according to Lemma 3.4, we have

$$f_\mu(z_k) - f_\mu(\tilde{z}) \geq \frac{1}{4(p+2)!M_\mu^{\frac{1}{p}}}\|\nabla f_\mu(\tilde{z})\|_2^{\frac{p+1}{p}} \quad (41)$$

At the same time,

$$f_\mu(z_k) - f_\mu(\tilde{z}) \leq f_\mu(z_k) - f_\mu(x_\mu^*) \leq \frac{\mu R_k^2}{2} \leq \tilde{\varepsilon} \quad (42)$$

by the stopping criterion of the algorithm. By combining (41) with (42) and from the choice of $\tilde{\varepsilon}$ we get that

$$\|\nabla f_\mu(\tilde{z})\|_2 \leq \frac{\varepsilon}{2}.$$

Since the right-hand side of (41) is non-negative, we can state that

$$f_\mu(\tilde{z}) \leq f_\mu(z_k). \quad (43)$$

From this and the definition of a strongly convex function, we have that

$$\frac{\mu}{2}\|\tilde{z} - x_\mu^*\|_2^2 \leq f_\mu(\tilde{z}) - f_\mu(x_\mu^*) \overset{(43)}{\leq} f_\mu(z_k) - f_\mu(x_\mu^*) \leq \frac{\mu R_k^2}{2} = \frac{\mu}{2}(R \cdot 2^{-k})^2 \leq \frac{\mu R^2}{2}.$$

Thus, $\|\tilde{z} - x_\mu^*\|_2 \leq R$. Hence, $\|\tilde{z} - x_0\|_2 \leq \|\tilde{z} - x_\mu^*\|_2 + \|x_\mu^* - x_0\|_2 \leq 2R$.

Finally, from our choice of μ

$$\|\nabla f(\tilde{z})\|_2 \leq \|\nabla f_\mu(\tilde{z})\|_2 + \mu\|\tilde{z} - x_0\|_2 \leq \frac{\varepsilon}{2} + \mu \cdot 2R \leq \varepsilon. \tag{44}$$

It remains to estimate the upper bound of the number of iterations of the Algorithm 1. Summing up the number of operations $\tilde{N}_i,\ i = 0, ..., k$ from (40), we obtain

$$\sum_{i=0}^{k} \tilde{N}_i \leq \sum_{i=0}^{k} \left[\left(\frac{8c_p M_p R_i^{p-1}}{\mu} \right)^{\frac{2}{3p+1}} + 1 \right] = \left(\frac{8c_p M_p R^{p-1}}{\mu} \right)^{\frac{2}{3p+1}} \sum_{i=0}^{k} 2^{\frac{-2i(p-1)}{3p+1}} + k$$

Again, as in Theorem 4.1, since $p \geq 2$, $\sum_{i=0}^{k} 2^{\frac{-2i(p-1)}{3p+1}}$ is a geometric progression with a common ratio lower than one. Therefore, we can upper bound its partial sum with its infinite sum.

$$\sum_{i=0}^{k} 2^{\frac{-2i(p-1)}{3p+1}} \leq \frac{2}{1 - 2^{\frac{-2(p-1)}{3p+1}}} \leq 2 \cdot 5 = 10.$$

$$\begin{aligned}\sum_{i=0}^{k} \tilde{N}_i &\leq \left(\frac{8c_p M_p R^{p-1}}{\mu} \right)^{\frac{2}{3p+1}} \sum_{i=0}^{k} 2^{\frac{-2i(p-1)}{3p+1}} + k \leq 10\left(\frac{8c_p M_p R^{p-1}}{\mu} \right)^{\frac{2}{3p+1}} + \frac{1}{2}\log_2 \frac{\mu R^2}{2\tilde{\varepsilon}} \\ &= O\left(\frac{M_p^{\frac{2}{3p+1}} R^{\frac{2p}{3p+1}}}{\varepsilon^{\frac{2}{3p+1}}} + \frac{1}{2}\log \frac{2^{\frac{p}{p-1}}(p+1)! M_p^{\frac{1}{p}}}{\varepsilon^{\frac{1}{p+1}}} \right). \end{aligned} \tag{45}$$

According to Theorem 3.1, the total number of oracle calls is within the $O\left(\ln(1/\varepsilon)\right)$ factor from the number of iterations of Algorithm 1. This completes the proof. □

If we omit the dominated factors in the result (45), we obtain the complexity bound

$$\tilde{O}\left(\frac{M_p R^p}{\varepsilon} \right)^{\frac{2}{3p+1}}. \tag{46}$$

Therefore, we can conclude that this bound coincides with the lower bound

$$\Omega\left(\frac{M_p R^p}{\varepsilon} \right)^{\frac{2}{3p+1}},$$

from [22] up to logarithmic and constant factors.

Remark 1. As a byproduct, Algorithm 3 can minimize functions f that are already strongly convex. Indeed, in this case, we deal with the objective f as we now deal with

the auxiliary objective f_μ: we apply Algorithm 1 by epochs to f and restart when the stopping criterion holds. Since in this case μ is not a regularization coefficient, but just a constant of strong convexity, we do not need to subsitute μ with $\varepsilon^2/(32\Delta_0)$ in (36) and with $\varepsilon/(4R)$ in (45). Thus, we get the following complexity estimations:

$$\tilde{O}\left(\frac{M_p \Delta_0^{\frac{p-1}{2}}}{\mu^{\frac{p+1}{2}}}\right)^{\frac{2}{3p+1}} \quad \text{and} \quad \tilde{O}\left(\frac{M_p R^{p-1}}{\mu}\right)^{\frac{2}{3p+1}}. \tag{47}$$

The main difference between Algorithms 2 and 3 is that in both cases, we use (19) to estimate the number of inner iterations of Algorithm 1, but in the first case we additionally use strong convexity to be able to upper bound argument residual with functional residual in (19): $R_k \leq \sqrt{(2/\mu)\Delta_k}$.

Remark 2. Let us now derive a complexity estimation for finding approximate solution to (1), using Algorithm 3. The idea is to apply Algorithm 3 to the dual problem and then use Proposition 2.1. To that end, we set the following equivalence between the notation of Section 2 and the notation of this section $\lambda \equiv z$, $\varphi(\lambda) \equiv f(z)$. We start by estimating the number of iterations of Algorithm 1 to fulfill the first condition of Proposition 2.1

$$-\langle \lambda_k, \nabla\varphi(\lambda_k)\rangle \leq \varepsilon_f. \tag{48}$$

Assume that after applying Algorithm 3, we obtain a point λ_k such that $\|\nabla\varphi(\lambda_k)\|_2 \leq \varepsilon$. Then,

$$-\langle \lambda_k, \nabla\varphi(\lambda_k)\rangle \leq \|\lambda_k\|_2 \|\nabla\varphi(\lambda_k)\|_2 \leq \varepsilon\|\lambda_k\|_2 \tag{49}$$

From the triangle inequality, (38), and (39), we have

$$\|\lambda_k\|_2 \leq \|\lambda_0\|_2 + \|\lambda_0 - \lambda_\mu^*\|_2 + \|\lambda_k - \lambda_\mu^*\|_2 \overset{(38),(39)}{\leq} \|\lambda_0\|_2 + 2R,$$

where we also used that from (39) $\|\lambda_k - \lambda_\mu^*\|_2 \leq R_k \leq R$. Since λ_0 is our choice (in particular, we can start our algorithm from $\lambda_0 = 0$), we can use this inequality to estimate $\|\lambda_k\|_2$. From the above and (49), we get

$$-\langle \lambda_k, \nabla\varphi(\lambda_k)\rangle \leq \varepsilon(2R + \|\lambda_0\|_2) = \varepsilon_f.$$

Thus, if we set $\varepsilon = \frac{\varepsilon_f}{2R+\|\lambda_0\|_2}$, we obtain that the first condition of Proposition 2.1 holds. If we set $\varepsilon = \varepsilon_{eq}$, we also obtain the second condition of this proposition. Setting $\lambda_0 = 0$ and $\varepsilon = \min\{\frac{\varepsilon_f}{2R}, \varepsilon_{eq}\}$, and applying the bound (46), we finally obtain the following complexity bound for finding an approximate solution to problem (8) in the sense of (10)

$$\tilde{O}\left(\max\left\{\left(\frac{M_p R^{p+1}}{\varepsilon_f}\right)^{\frac{2}{3p+1}}, \left(\frac{M_p R^p}{\varepsilon_{eq}}\right)^{\frac{2}{3p+1}}\right\}\right).$$

While Algorithms 2 and 3 are shown to be near-optimal, the price of optimality of the algorithm is high. We believe that pointing out this price of optimality can lead to

future research in computationally tractable approaches. Specifically, the algorithms require a line search process, which adds logarithmic terms to the complexity. Moreover, they depend on restart techniques and regularization, whose parameters depend on the desired accuracy ε and other parameters such as R, which are assumed to be known.

5. Primal-dual accelerated tensor method

In Section 4, we considered methods, which search for an approximate stationary point of the dual problem, and then reconstruct an approximate solution to the primal problem. Another approach to tackle (8) is via primal-dual methods. The main idea of these methods is to solve both dual and primal problems until both duality gap $|f(x(\lambda_k))+\varphi(\lambda_k)|$ and equality constraint residual of primal variable $\|Ax(\lambda_k)-b\|_2$ are lower than some accuracy ε.

In this section, we compare these two approaches theoretically and numerically. Hence, in this section, we propose an accelerated primal-dual tensor method (Algorithm 4) and provide its theoretical comparison with Algorithms 2 and 3 in Remark 3. Our proposed method uses the framework of estimating sequences [34], where in each step it solves high-order optimization Problem (5) for the dual function φ.

First, recall formulation of the dual problem for (8)

$$\min_{\lambda\in H^*}\left\{\varphi(\lambda) := \langle\lambda, b\rangle + \max_{x\in Q}\left(-f(x) - \langle A^\intercal\lambda, x\rangle\right)\right\}. \tag{50}$$

We have already mentioned it in Section 2. We assume the dual function φ has M_p-Lipschitz p-th order derivative (see, e.g., Section 2). From the weak duality, the following inequality follows

$$f(x^*) \geq -\varphi(\lambda^*), \tag{51}$$

where $f(x^*)$ and $\varphi(\lambda^*)$ are the optimal function values in (8) and (9) respectively.

Assume the dual problem (50) has a solution λ^* (which holds, e.g., when the strong duality holds), and there exists some $R > 0$ such that

$$\|\lambda^*\|_2 \leq R < +\infty. \tag{52}$$

It is worth noting that the quantity R will be used only in the convergence analysis but not in the algorithm itself.

To solve the dual Problem (50), we introduce the Primal-Dual Accelerated Tensor Method (Algorithm 4). To prove the main result about the convergence of Algorithm 4, we need the following auxiliary lemmas.

Lemma 5.1 (Corollary 1 in [37]). *For any $\lambda \in H^*$ and $M \geq M_p$ we have*

$$\langle\nabla\varphi(T^{\varphi}_{p,M}(\lambda)), \lambda - T^{\varphi}_{p,M}(\lambda)\rangle \geq \tfrac{c(p)}{M}[M^2 - M_p^2]^{\frac{p-1}{2p}}\|\nabla\varphi(T^{\varphi}_{p,M}(\lambda))\|_2^{\frac{p+1}{p}}, \tag{53}$$

where $c(p) = \frac{p}{p-1}\left[\frac{p-1}{p+1}\right]^{\frac{1-p}{2p}}[(p+1)!]^{\frac{1}{p}}$.

Algorithm 4 Primal-Dual Accelerated Tensor Method

Require: ε_f, ε_{eq}, $M > M_p$.

Set $k = 0$, $\lambda_0 = 0$, $\psi_0(\lambda) = \frac{C}{(p+1)!}\|\lambda - \lambda_0\|_2^{p+1}$, where $C = \frac{p}{2}\sqrt{\frac{p+1}{p-1}(M^2 - M_p^2)}$.

repeat

  Compute $v_k = \operatorname{argmin}_\lambda \psi_k(\lambda)$.

  $A_k = \left[\frac{(p-1)(M^2-M_p^2)}{4(p+1)p^2M^2}\right]^{\frac{p}{2}} \left(\frac{k}{p+1}\right)^{p+1}$, $a_k = A_{k+1} - A_k$.

  $y_k = \frac{A_k}{A_{k+1}}\lambda_k + \frac{a_k}{A_{k+1}}v_k$.

  Compute $\lambda_{k+1} = T^{\varphi}_{p,M}(y_k)$.

$$\psi_{k+1}(\lambda) = \psi_k(\lambda) + (A_{k+1} - A_k)\left[\varphi(\lambda_{k+1}) + \langle \nabla\varphi(\lambda_{k+1}), \lambda - \lambda_{k+1}\rangle\right].$$

$$\hat{x}_{k+1} = \frac{1}{A_{k+1}}\sum_{i=0}^{k} a_i x(\lambda_{i+1}) = \frac{a_k x(\lambda_{k+1}) + A_k\hat{x}_k}{A_{k+1}}$$

  Set $k = k + 1$.

until $|f(\hat{x}_k) + \varphi(\lambda_k)| \le \varepsilon_f$, $\|A\hat{x}_k - b\|_2 \le \varepsilon_{eq}$.

return $\hat{x}_k$, λ_k.

Lemma 5.2 (Lemma 2 in [33]). *Let $\sigma > 0$ be some constant. Then, for any $h \in E$ and $s \in E$, we have*

$$\langle s, h\rangle + \frac{1}{p}\sigma\|h\|_2^p \ge -\frac{p-1}{p}\left(\frac{1}{\sigma}\right)^{\frac{1}{p-1}}\|s\|_2^{\frac{p}{p-1}}. \tag{54}$$

Let us introduce the following estimating functions, which are recursively updated as

$$\forall k \ge 0 \Rightarrow \psi_{k+1}(\lambda) = \psi_k(\lambda) + a_k\left[\varphi(\lambda_{k+1}) + \langle \nabla\varphi(\lambda_{k+1}), \lambda - \lambda_{k+1}\rangle\right] \tag{55}$$

with $\psi_0(\lambda) = \frac{C}{(p+1)!}\|\lambda - \lambda_0\|_2^{p+1}$, where $C = \frac{p}{2}\sqrt{\frac{p+1}{p-1}(M^2 - M_p^2)}$.

Theorem 5.3. *If sequence $\{\lambda_k\}_{k=0}^{\infty}$ is generated by Algorithm 4, then for all $k \ge 0$ we have*

$$A_k\varphi(\lambda_k) \le \min_{\lambda \in H^*}\psi_k(\lambda). \tag{56}$$

Proof. Let us prove the relation (56) by induction over k. Since $A_0 = 0$, for $k = 0$ we obtain:

$$0 = A_0\varphi(\lambda_0) \le \min_{\lambda\in H^*}\frac{C}{(p+1)!}\|\lambda - \lambda_0\|_2^{p+1} = 0.$$

Assume that (56) is true for some $k > 0$. Denote

$$\psi_k(\lambda) \equiv l_k(\lambda) + \frac{C}{(p+1)!}\|\lambda - \lambda_0\|_2^{p+1} \quad k \ge 0,$$

where $l_0(\lambda) \equiv 0$.

Using Lemma 4 in [33] we obtain that

$$\begin{aligned}\psi_k(\lambda) &\geq \min_{\lambda\in H^*}\psi_k(\lambda) + \tfrac{C}{(p+1)!}\cdot\left(\tfrac{1}{2}\right)^{p-1}\|\lambda - v_k\|_2^{p+1}\\ &\geq A_k\varphi(\lambda_k) + \tfrac{C}{(p+1)!}\cdot\left(\tfrac{1}{2}\right)^{p-1}\|\lambda - v_k\|_2^{p+1}.\end{aligned}$$

Denote $\sigma_{p+1} = \frac{C}{p!}\left(\frac{1}{2}\right)^{p-1}$. Then, from this inequality and $\varphi(\lambda_k) - \varphi(\lambda_{k+1}) \geq \langle\nabla\varphi(\lambda_{k+1}, \lambda_k - \lambda_{k+1}\rangle$, we get

$$\begin{aligned}&\psi^*_{k+1} = \min_{\lambda\in H^*}\{\psi_k(\lambda) + a_k[\varphi(\lambda_{k+1}) + \langle\nabla\varphi(\lambda_{k+1}), \lambda - \lambda_{k+1}\rangle]\}\\ \geq\ &\min_{\lambda\in H^*}\Big\{A_k\varphi(\lambda_k) + \tfrac{\sigma_{p+1}}{(p+1)}\|\lambda - v_k\|_2^{p+1} + a_k[\varphi(\lambda_{k+1}) + \langle\nabla\varphi(\lambda_{k+1}), \lambda - \lambda_{k+1}\rangle]\Big\}\\ \geq\ &\min_{\lambda\in H^*}\Big\{A_{k+1}\varphi(\lambda_{k+1}) + A_k\langle\nabla\varphi(\lambda_{k+1}), \lambda_k - \lambda_{k+1}\rangle\\ +\ &a_k\langle\nabla\varphi(\lambda_{k+1}), \lambda - \lambda_{k+1}\rangle + \tfrac{\sigma_{p+1}}{(p+1)}\|\lambda - v_k\|_2^{p+1}\Big\}\end{aligned} \tag{57}$$

Note, that $y_k = \frac{A_k}{A_{k+1}}\lambda_k + \frac{a_k}{A_{k+1}}v_k$. Hence, $A_k\lambda_k = A_{k+1}y_k - a_kv_k$, and

$$A_k\left\langle\nabla\varphi(\lambda_{k+1}), \lambda_k - \lambda_{k+1}\right\rangle = \left\langle\nabla\varphi(\lambda_{k+1}), A_{k+1}y_k - a_kv_k - A_k\lambda_{k+1}\right\rangle.$$

Thus, we can rewrite (57) as follows

$$\begin{aligned}\psi^*_{k+1} &\geq \min_{\lambda\in H^*}\Big\{A_{k+1}\varphi(\lambda_{k+1}) + \langle\nabla\varphi(\lambda_{k+1}), A_{k+1}y_k - a_kv_k - A_k\lambda_{k+1}\rangle\\ &+ a_k\langle\nabla\varphi(\lambda_{k+1}), \lambda - \lambda_{k+1}\rangle + \tfrac{\sigma_{p+1}}{(p+1)}\|\lambda - v_k\|_2^{p+1}\Big\}\\ &= \min_{\lambda\in H^*}\Big\{A_{k+1}\varphi(\lambda_{k+1}) + A_{k+1}\left\langle\nabla\varphi(\lambda_{k+1}), y_k - \lambda_{k+1}\right\rangle\\ &+ a_k\langle\nabla\varphi(\lambda_{k+1}), \lambda - v_k\rangle + \tfrac{\sigma_{p+1}}{(p+1)}\|\lambda - v_k\|_2^{p+1}\Big\}\end{aligned} \tag{58}$$

Further, if we choose $M \geq M_p$, then by inequality (53) we have

$$\langle\nabla\varphi(\lambda_{k+1}), \lambda - \lambda_{k+1}\rangle \geq \tfrac{c(p)}{M}[M^2 - M_p^2]^{\frac{p-1}{2p}}\|\nabla\varphi(\lambda_{k+1})\|_2^{\frac{p+1}{p}}. \tag{59}$$

If we apply (59) to (58), we get

$$\begin{aligned}\psi^*_{k+1} \geq \min_{\lambda\in H^*}\Big\{&A_{k+1}\varphi(\lambda_{k+1}) + A_{k+1}\tfrac{c(p)}{M}[M^2 - M_p^2]^{\frac{p-1}{2p}}\|\nabla\varphi(\lambda_{k+1})\|_2^{\frac{p+1}{p}}\\ &+ a_k\langle\nabla\varphi(\lambda_{k+1}), \lambda - v_k\rangle + \tfrac{\sigma_{p+1}}{(p+1)}\|\lambda - v_k\|_2^{p+1}\Big\}.\end{aligned}$$

Now, denote everything on the right-hand side except $A_{k+1}\varphi(\lambda_{k+1})$ as $\zeta(\lambda)$:

$$\zeta(\lambda) \equiv A_{k+1}\tfrac{c(p)}{M}[M^2 - M_p^2]^{\frac{p-1}{2p}}\|\nabla\varphi(\lambda_{k+1})\|_2^{\frac{p+1}{p}} + a_k\langle\nabla\varphi(\lambda_{k+1}), \lambda - v_k\rangle + \frac{\sigma_{p+1}}{p+1}\|\lambda - v_k\|_2^{p+1}.$$

Thus,

$$\psi^*_{k+1} \geq \min_{\lambda \in H^*} \Big\{ A_{k+1}\varphi(\lambda_{k+1}) + \zeta(\lambda) \Big\}.$$

To prove (56), we need to have $\zeta(\lambda) \geq 0$. Using (54), we get

$$\zeta(\lambda) \geq A_{k+1}\tfrac{c(p)}{M}[M^2 - M_p^2]^{\frac{p-1}{2p}} \|\nabla\varphi(\lambda_{k+1})\|_2^{\frac{p+1}{p}} - \frac{p}{p+1}\left(\frac{1}{\sigma_{p+1}}\right)^{\frac{1}{p}} a_k^{\frac{p+1}{p}} \|\nabla\varphi(\lambda_{k+1})\|_2^{\frac{p+1}{p}}.$$

Therefore, to have $\zeta(\lambda) \geq 0$ we need

$$A_{k+1}\tfrac{c(p)}{M}[M^2 - M_p^2]^{\frac{p-1}{2p}} \geq \frac{p}{p+1}\left(\frac{1}{\sigma_{p+1}}\right)^{\frac{1}{p}} a_k^{\frac{p+1}{p}}.$$

Next we substitute in this inequality the values of $c(p)$ and σ_{p+1} and after all the constellations we get

$$A_{k+1}\sqrt{1 - \frac{M_p^2}{M^2}}\left(\frac{C^2}{M^2 - M_p^2}\right)^{\frac{1}{2p}} \geq 2a_k^{\frac{p+1}{p}}\left(\frac{p}{2}\sqrt{\frac{p+1}{p-1}}\right)^{\frac{1}{p}}\sqrt{\frac{p+1}{p-1}}.$$

Finally, from our choice of C, we get

$$A_{k+1} \geq 2\sqrt{\tfrac{(p+1)M^2}{(p-1)(M^2-M_p^2)}}\, a_k^{\frac{p+1}{p}}. \tag{60}$$

And since for $k \geq 0$

$$A_k = \left[\tfrac{(p-1)(M^2-M_p^2)}{4(p+1)M^2}\right]^{\frac{p}{2}}\left(\tfrac{k}{p+1}\right)^{p+1}, \quad a_k = A_{k+1} - A_k.$$

inequality (60) holds. It is described in more detail in [37] (everything from eq. (3.8) to eq. (3.11)). Eventually,

$$\psi^*_{k+1} \geq \min_{\lambda \in H^*} \Big\{ A_{k+1}\varphi(\lambda_{k+1}) + \zeta(\lambda) \Big\} \geq A_{k+1}\varphi(\lambda_{k+1}),$$

that completes the induction argument □

We can now estimate the proposed algorithm's complexity. Consider the set $\Lambda_R = \{\lambda : \|\lambda\|_2 \leq 2R\}$ where R is given in (52). From the Theorem 5.3 and since $\lambda_0 = 0$ we

obtain

$$\begin{aligned} A_k\varphi(\lambda_k) &\leq \min_{\lambda}\left\{\sum_{i=0}^{k-1} a_i\left[\varphi(\lambda_{i+1})+\langle\nabla\varphi(\lambda_{i+1}),\lambda-\lambda_{i+1}\rangle\right]+\frac{C}{(p+1)!}\|\lambda\|_2^{p+1}\right\} \\ &\leq \min_{\lambda\in\Lambda_R}\left\{\sum_{i=0}^{k-1} a_i\left[\varphi(\lambda_{i+1})+\langle\nabla\varphi(\lambda_{i+1}),\lambda-\lambda_{i+1}\rangle\right]+\frac{C}{(p+1)!}\|\lambda\|_2^{p+1}\right\} \\ &\leq \min_{\lambda\in\Lambda_R}\left\{\sum_{i=0}^{k-1} a_i\left[\varphi(\lambda_{i+1})+\langle\nabla\varphi(\lambda_{i+1}),\lambda-\lambda_{i+1}\rangle\right]\right\}+\frac{C(2R)^{p+1}}{(p+1)!}. \end{aligned} \tag{61}$$

On the other hand, from the definition (50) of $\varphi(\lambda)$, we have

$$\begin{aligned} \varphi(\lambda_i) &= \langle\lambda_i,b\rangle+\max_{x\in Q}\left(-f(x)-\langle A^\intercal\lambda_i,x\rangle\right) \\ &= \langle\lambda_i,b\rangle-f(x(\lambda_i))-\langle A^\intercal\lambda_i,x(\lambda_i)\rangle. \end{aligned}$$

And since $\nabla\varphi(\lambda)=b-Ax(\lambda)$, we obtain

$$\begin{aligned} \varphi(\lambda_i)-\langle\nabla\varphi(\lambda_i),\lambda_i\rangle &= \langle\lambda_i,b\rangle-f(x(\lambda_i))-\langle A^\intercal\lambda_i,x(\lambda_i)\rangle \\ &\quad-\langle b-Ax(\lambda_i),\lambda_i\rangle=-f(x(\lambda_i)). \end{aligned}$$

Summing these inequalities from $i=0$ to $i=k-1$ with the weights $\{\alpha_i\}_{i=0,\dots k-1}$, we get, using the convexity of f

$$\begin{aligned} &\sum_{i=0}^{k-1}\alpha_i\left(\varphi(\lambda_{i+1})+\langle\nabla\varphi(\lambda_{i+1}),\lambda-\lambda_{i+1}\rangle\right) \\ &=-\sum_{i=0}^{k-1}\alpha_i f(x(\lambda_{i+1}))+\sum_{i=0}^{k-1}\alpha_i\langle b-Ax(\lambda_{i+1}),\lambda\rangle \\ &\leq -A_k f(\hat{x}_k)+A_k\langle b-A\hat{x}_k,\lambda\rangle, \end{aligned}$$

where $\hat{x}_k=\frac{1}{A_k}\sum\limits_{i=0}^{k-1}a_i x(\lambda_{i+1})$. Substituting this inequality to (61), we obtain

$$A_k\varphi(\lambda_k)\leq -A_k f(\hat{x}_k)+A_k\min_{\lambda\in\Lambda_R}\{\langle b-A\hat{x}_k,\lambda\rangle\}+\frac{C(2R)^{p+1}}{(p+1)!}.$$

Finally, since

$$\max_{\lambda\in\Lambda_R}\{\langle A\hat{x}_k-b,\lambda\rangle\}=2R\|A\hat{x}_k-b\|_2,$$

we obtain

$$\varphi(\lambda_k)+f(\hat{x}_k)+2R\|A\hat{x}_k-b\|_2\leq\frac{C(2R)^{p+1}}{A_k(p+1)!}. \tag{62}$$

Since λ^* is an optimal solution of dual problem (50), we have, for any $x \in Q$

$$f(x^*) \leq f(x) + \langle \lambda^*, Ax - b \rangle.$$

Using the assumption (52) we get

$$f(\hat{x}_k) \geq f(x^*) - R\|A\hat{x}_k - b\|_2. \tag{63}$$

Hence,

$$\begin{aligned} \varphi(\lambda_k) + f(\hat{x}_k) &= \varphi(\lambda_k) - \varphi(\lambda^*) + \varphi(\lambda^*) + f(x^*) - f(x^*) + f(\hat{x}_k) \\ &\overset{(51)}{\geq} -f(x^*) + f(\hat{x}_k) \overset{(63)}{\geq} -R\|A\hat{x}_k - b\|_2. \end{aligned} \tag{64}$$

This and (62) give

$$R\|A\hat{x}_k - b\|_2 \leq \frac{C(2R)^{p+1}}{A_k(p+1)!}. \tag{65}$$

Hence, we obtain

$$-\frac{C(2R)^{p+1}}{A_k(p+1)!} \overset{(64),(65)}{\leq} \varphi(\lambda_k) + f(\hat{x}_k) \overset{(62)}{\leq} \frac{C(2R)^{p+1}}{A_k(p+1)!}. \tag{66}$$

Combining (65) and (66), we conclude

$$R\|A\hat{x}_k - b\|_2 \leq \frac{C(2R)^{p+1}}{A_k(p+1)!}, \quad |\varphi(\lambda_k) + f(\hat{x}_k)| \leq \frac{C(2R)^{p+1}}{A_k(p+1)!}.$$

Finally, if we put the value of A_k, defined in Algorithm 4, we will get the total complexity of the Algorithm 4. Therefore, we have just proved the following theorem.

Theorem 5.4. *Assume the function φ from (50) is convex, p times differentiable on $\mathbb{R}^m$ with M_p-Lipschitz p-th derivative. Additionally, if $\lambda^* = \arg\min_{\lambda \in H^*} \varphi(\lambda)$, assume $\exists R > 0 : \|\lambda^*\|_2 \leq R \leq \infty$. Let Algorithm 4 be run for k steps with starting point $\lambda_0 = v_0 = z_0 = 0$. Denote $\hat{x}_k = \frac{1}{A_k} \sum\limits_{i=0}^{k-1} a_i x(\lambda_{i+1})$. Then*

$$\begin{aligned} \|A\hat{x}_k - b\|_2 &\leq \frac{C_1 R^p}{k^{p+1}}, \\ |\varphi(\lambda_k) + f(\hat{x}_k)| &\leq \frac{C_1 R^{p+1}}{k^{p+1}}. \end{aligned}$$

Here $C_1 = \frac{4^p M^p}{(p-1)!} \sqrt{\frac{(p+1)^{3p+3}}{(M^2 - M_p^2)^{p-1}(p-1)^{p+1}}}$.

The result of the above Theorem can be written in terms of complexity in the following way. Assume that the goal is to find an approximate solution that satisfies inequalities (10). Then, Theorem 5.4 states that such a point can be found in a number

of iterations not exceeding

$$O\left(\max\left\{\left(\frac{M_p R^{p+1}}{\varepsilon_f}\right)^{\frac{1}{p+1}}, \left(\frac{M_p R^p}{\varepsilon_{eq}}\right)^{\frac{1}{p+1}}\right\}\right).$$

It is an open question whether it is possible to obtain a primal-dual tensor method with complexity bounds that depend on $\varepsilon^{-2/(3p+1)}$ rather than $\varepsilon^{-1/(p+1)}$.

Remark 3. Let us now discuss Algorithm 4 compared to Algorithms 2 and 3. On the one hand, complexity $O(\varepsilon^{-1/(p+1)})$ of the former has *asymptotically* worse dependence on ε than the complexity bound $\tilde{O}(\varepsilon^{-2/(3p+1)})$ of the latter. On the other hand, the difference in the power of ε is quite small, and the second bound has an additional logarithmic multiplier. Thus, the bound for Algorithms 2 and 3 may be only slightly better than the bound for Algorithm 4. Further, Algorithm 4 is a direct algorithm that does not use regularizations and restarts. Unlike it, Algorithms 2 and 3 use a regularization that may be so small that it will cause some numerical instabilities. In Section 6, we compare both approaches numerically. At the same time, Algorithms 2 and 3 are interesting not only in application to problem (8). These methods achieve nearly-optimal complexity bounds for finding approximate stationary points of convex functions, nearly closing the theoretical gap. Some other motivations for developing efficient methods for finding stationary points can be found in [22]. In particular, the norm of the gradient is a natural and computable measure of optimality.

Remark 4. Let us discuss a possible extension of the proposed methods. One straightforward generalization is a near-optimal method for minimizing the norm of objective with Hölder-continuous gradient, i.e., for some $\nu \in [0, 1]$ satisfying

$$\|\nabla^p f(x) - \nabla^p f(y)\|_2 \leq M_{p,\nu}\|x - y\|_2^\nu, x, y \in \mathbb{R}^n.$$

The idea is to combine the near-optimal tensor method for minimization of functions with Hölder-continuous p-th derivatives [42] with Lemma 5.2 in [22] for general ν. This approach allows obtaining complexity bounds which, up to logarithmic and constant factors, coincide with the lower bounds in [22].

Another possible extension is an inexact solution of the auxiliary subproblems and adaptation to the constant $M_{p,\nu}$ [22]. Importantly, the basic Algorithm 1 is adaptive to M_p. Nevertheless, to apply the regularization technique with parameter μ, we need to know M_p. Thus, it is desirable to overcome this drawback.

Finally, in our Algorithm 4 we use Nesterov acceleration based on the estimating sequence technique (see [34, 37]). It is still an open question whether we can obtain a better high-order primal-dual method using the Monteiro-Svaiter acceleration [32] or optimal tensor method [28].

6. Numerical analysis

This section presents several simulations for proposed methods. Particularly, we implement Algorithm 2 for the logistic regression problem on both synthetic and real data sets. Also, we show the performance of Algorithm 2 on a family of functions recently described as difficult for all tensor methods [37]. We focus on the case where $p = 3$ for which we have efficient methods for solving the auxiliary subproblem [37, Section

5]. Finally, we present the performance results for the entropy regularized optimal transport and minimal mutual information problems.

6.1. Logistic Regression

For the logistic regression problem, we are given a set of d data pairs $\{y_i, w_i\}$ for $1 \leq i \leq d$, where $y_i \in \{1, -1\}$ is the class label of object i, and $w_i \in \mathbb{R}^n$ is the set of features of object i. After the dimension and number of data points are set. The optimal point is generated as x^* composed in each dimension as samples from a uniform distribution in the range $[-1, 1]$. Each dimension per data sample is also generated as samples from a uniform distribution in the range $[-1, 1]$ with the last feature set to 1 for all data points. The label is generated as the sign of the products of the features and x^*. Finally, labels are flipped with a probability of 0.01. We are interested in finding a vector x that solves the following optimization problem

$$\min_{x \in \mathbb{R}^n} \frac{1}{d} \sum_{i=1}^{d} \ln\Big(1 + \exp\big(-y_i \langle w_i, x \rangle\big)\Big). \tag{67}$$

Figure 1 shows the gradient norm of the logistic regression function at the points generated by Algorithm 2. Initially, we show the results for synthetic data where $d = 100$ and $n = 10$. We focus on showing the results for different values of ε. Here by Iterations we mean the number of iterations of Algorithm 1 inside Algorithm 2, line 3. For implementation simplicity, in addition to stopping criterion $A_{N_k} \geq \frac{2}{\mu}$, if the gradient is no longer decreasing, we apply the restarting of Algorithm 1 after 500 iterations.

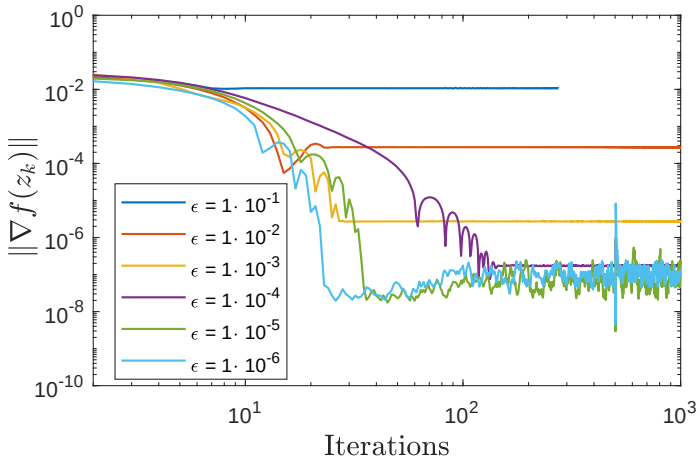


Figure 1. Gradient norm at the iterations generated by Algorithm 2 on synthetic data for various values of ε.

Figure 2 shows the gradient norm of the logistic regression function at the points generated by Algorithm 2. In this case, we use the `Mushroom`, `A9A`, `Covertype` and `IJCNN1` datasets from [16] with a fixed value of $\varepsilon = 1 \cdot 10^{-5}$.

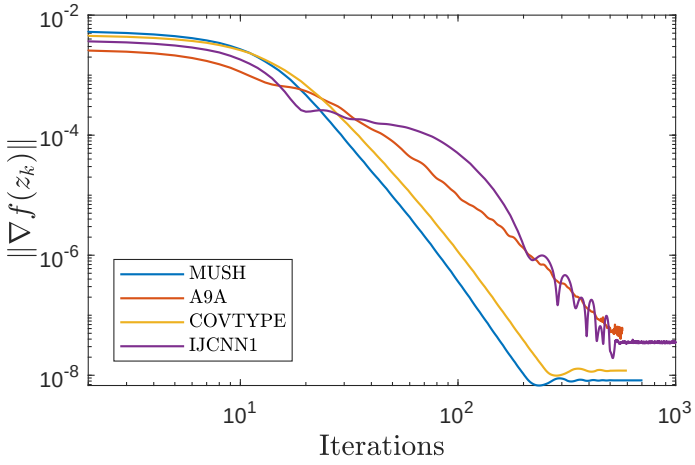


Figure 2. Gradient norm at the iterations generated by Algorithm 2 on real data sets from [16] with $\varepsilon = 1 \cdot 10^{-5}$.

6.2. A family of difficult functions

Next, we analyze the performance of the proposed algorithm on a universal parametric family of objective functions, which are difficult for all tensor methods [22, 37] defined as

$$f_m(x) = \eta_{p+1}(A_m x) - x_1, \tag{68}$$

where, for integer parameter $p \geq 1$, $\eta_{p+1}(x) = \frac{1}{p+1}\sum_{i=1}^{n} |x_i|^{p+1}$, $2 \leq m \leq n$, $x \in \mathbb{R}^n$, A_m is the $n \times n$ block diagonal matrix:

$$A_m = \begin{pmatrix} U_m & 0 \\ 0 & I_{n-m} \end{pmatrix}, \quad \text{with} \quad U_m = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \tag{69}$$

and I_n is the identity $n \times n$-matrix. For a detailed description of the high-order derivatives of this class of functions and its optimality properties, see [37].

Finally, Figure 3 shows the performance results of Algorithm 2 on the family of functions in (68) with $p = 3$ and various values of parameters $m = n$ with $\varepsilon = 1 \cdot 10^{-5}$.

6.3. The entropy regularized optimal transport problem

We now go back to the entropy-regularized optimal transport problem in (11) and present some numerical experiments of the proposed method applied to its dual problem in (12).

$$\phi(\lambda) = \mathbf{smax}_\gamma(A^\intercal \lambda - M) + \langle \lambda, b \rangle,$$

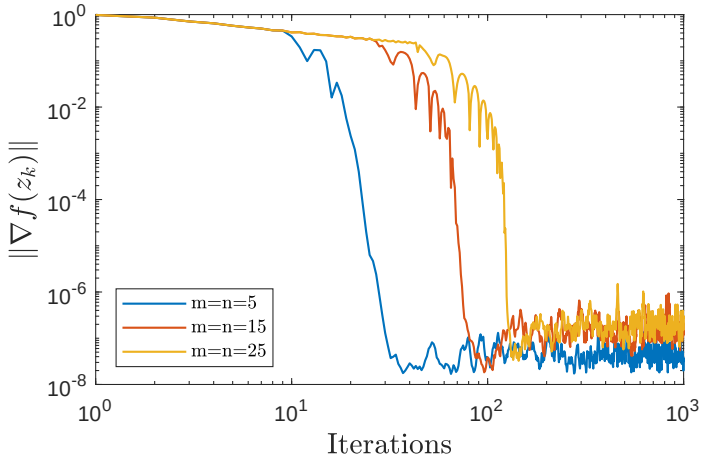


Figure 3. Gradient norm at the iterations generated by Algorithm 1 on the family of functions in (68) with $p = 3$ and various values of parameters $m = n$ with $\varepsilon = 1 \cdot 10^{-5}$.

Initially, we recall some properties of the Softmax function, which will be useful for the implementation; for a complete analysis of such properties, see [9].

Proposition 6.1. *Consider the softmax function in* (16)*, and define the function* $f(x) = \boldsymbol{smax}_\gamma(Ax - b) - \langle \lambda, b \rangle$*. Then the following properties hold:*

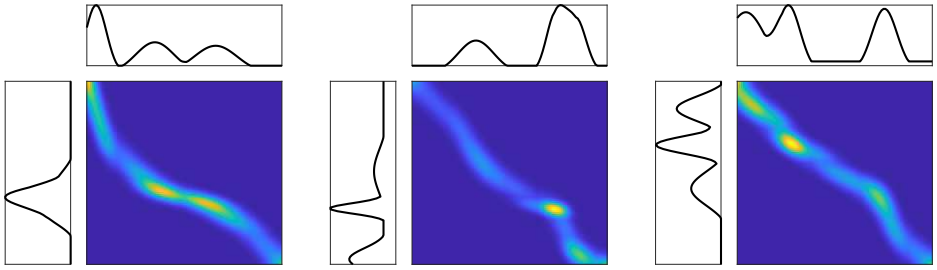
$$\nabla \boldsymbol{smax}_\mu(x)_i = \exp\left(\frac{x_i}{\mu}\right) \Big/ \left(\sum_i \exp\left(\frac{x_i}{\mu}\right)\right)$$
$$\nabla^2 \boldsymbol{smax}_\mu(x) = \frac{1}{\mu}\big(diag\big(\nabla \boldsymbol{smax}_\mu(x)\big) - \nabla \boldsymbol{smax}_\mu(x) \nabla \boldsymbol{smax}_\mu(x)^\intercal\big)$$
$$\nabla^3 \boldsymbol{smax}_\mu(x)[h,h] = \frac{1}{\mu}\big(\nabla^2 \boldsymbol{smax}_\mu(x)[h^2] - 2\langle \nabla \boldsymbol{smax}_\mu(x), h\rangle \nabla^2 \boldsymbol{smax}_\mu(x)[h]\big),$$

and

$$\nabla f(x) = A^\intercal \nabla \boldsymbol{smax}_\mu(Ax - b)$$
$$\nabla^2 f(x) = A^\intercal \nabla^2 \boldsymbol{smax}_\mu(Ax - b) A$$
$$\nabla^3 f(x)[h,h] = A^\intercal \nabla^3 \boldsymbol{smax}_\mu(Ax - b)[Ah, Ah].$$

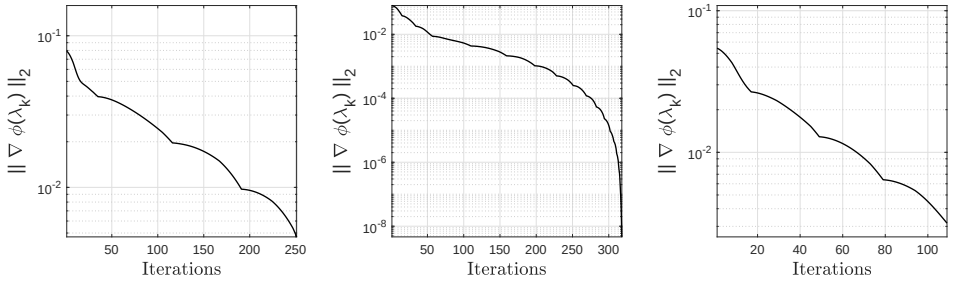
6.3.1. Discrete probability distributions

Next, we present the numerical results for the computation of the optimal (entropy-regularized) transport plan between two discrete probability distributions using the near-optimal third-order method in Algorithm 2. We construct two discrete distributions as the mixture of three randomly generated Gaussian distributions, each on bounded support $[-5, 5]$ with $n = 100$. We select the regularization parameter to $\gamma = 0.1$, which is common for these applications. Figure 4 shows three examples of the resulting transport plan obtained by Algorithm 2 for three different pairs of distributions, and the corresponding distributions are shown as the marginals of the transport plan. Figure 5 shows the corresponding norms of the gradients, evaluated at each iteration of Algorithm 2 for the three problems shown in Figure 4.



(a) (b) (c)

Figure 4. Three separate examples of the resulting transport plan obtained by Algorithm 2. The two distributions are shown on the left and top of the transport plan.



(a) (b) (c)

Figure 5. Norm of the gradient at each iteration of Algorithm 2 for the three problems shown in Figure 4.

6.3.2. MNIST

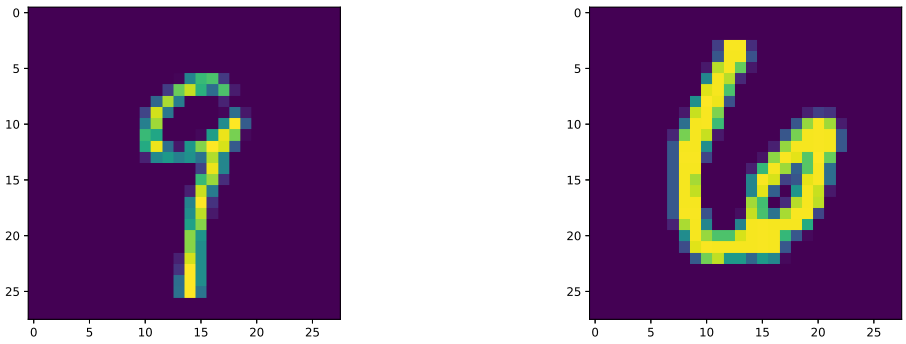
Finally, we provide the results of experiments for transportation plan computation between two MNIST images of handwritten digits. Here we compare the results of Algorithm 3 (GN) and Algorithm 4 (PDATM). These methods represent two approaches to tackle this problem: gradient norm minimization of dual function and primal-dual method. As we mentioned earlier, comparing these two approaches is our main motivation in this paper. Additionally, we provide the results of the Algorithm 1 (ATD) applied to dual function. In detail, we use it to minimize dual function until we achieve the prescribed ε-approximate solution for constraint and duality gap. In other words, we use it as a primal-dual method.

In this experiment, we take the images presented in Figure 6 as initial histograms. The size of each picture is 28×28 pixels. We reshape these images to vectors of size $n = 28^2 = 784$.

To perform the inner "tensor" step (5) inside each of the considered algorithms, we use the method developed in [37]. We use its inexact modification from [40] and look for the points from the following neighborhood:

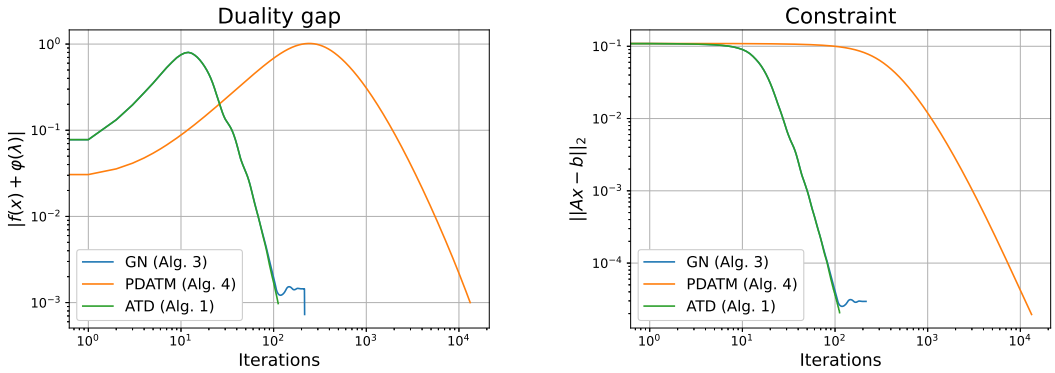
$$\mathcal{N}^{\alpha}_{p,M}(x) \equiv \{T \in \mathbb{R}^n : \|\nabla \Omega_{x,p,M}(T)\|_2 \leq \alpha \|\nabla f(T)\|_2\}, \tag{70}$$

where we choose the same size of the neighborhood as in [40]: $\alpha = \frac{1}{2p} = \frac{1}{6}$. Thus, we run our inner-problem method until we reach the point $T \in \mathbb{R}^n$, inside the set defined in (70).



(a) (b)

Figure 6. Initial and target images for optimal transport problem



(a) (b)

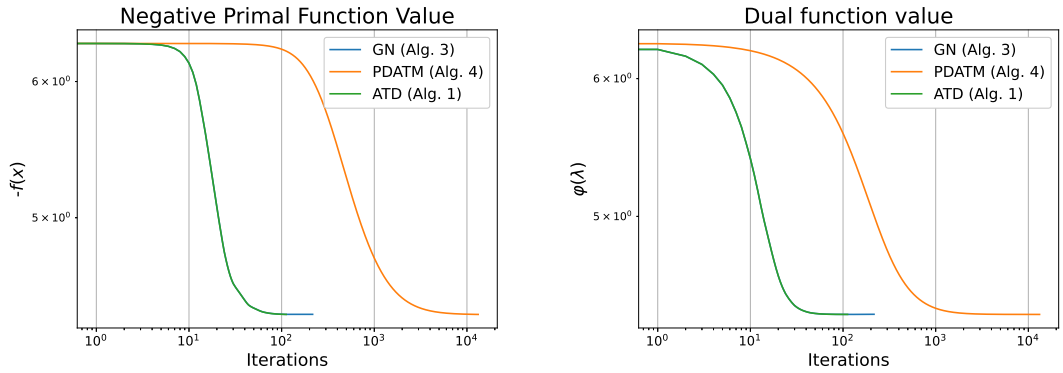
Figure 7. Duality gap and equality constraint convergence for Algorithms 3, 4 and 1.

We select the regularization parameter $\gamma = 0.5$ and Lipschitz constant $M_p = 0.5$. We want to emphasize that we do not use theoretical estimation of M_p from Proposition 2.2 because it gives too pessimistic upper bound, which results in too slow convergence. We start from zero and stop the optimization process for Algorithms 4 and 1 when both constraint and duality gap become smaller or equal than $\varepsilon = 0.001$. For Algorithm 3, we stop when the norm of the gradient of dual function is less or equal than $\min\{\varepsilon; \frac{\varepsilon}{2R}\}$. To choose R, we use the following lemma.

Lemma 6.2 (Lemma 11 in [24]). *Let $M \in \mathbb{R}_+^{n\times n}$ be a transportation matrix, $p, q \in \Sigma_n$ be two histograms. Then, there exists a solution (ξ^*, η^*) of (12) such that*

$$\|(\xi^*, \eta^*)\|_2 \leq R := \sqrt{N/2}\left(\|M\|_\infty - \frac{\gamma}{2}\ln\min_{i,j}\{p_i, q_j\}\right).$$

In Figure 7a, we show the duality gap convergence, and in Figure 7b, we show the convergence results for equality constraints. Additionally, we provide the results for values of $f(x)$ and $\varphi(\lambda)$ on Figure 8. In Figure 8a, we show a negative value of $f(x)$ since otherwise, we could not plot it in a log scale.



(a) (b)

Figure 8. Negative primal and dual function values for Algorithms 3 and 4

At first, we compare the results of Algorithm 3 and Algorithm 4, and then we discuss the performance of Algorithm 1. All these figures show that Algorithm 3 outperforms Algorithm 4 in convergence speed. The little spikes on Figures 7a and 7b at the end of the blue line are the result of restarts of Algorithm 1 inside Algorithm 3. The sharp drop at the end of the blue line is due to the final tensor step in Line 6 of Algorithm 3.

We mentioned earlier, and it is noticeable from pseudocode, that Algorithm 4 is a *direct* method, which means that it does not use restarts, regularization, or binary search. These modifications introduce difficulties in implementing the method and may slow it down. For example, restarts usually give a too-pessimistic upper bound for the number of iterations of the inner method. Nevertheless, the Primal-Dual method needs only Lipschitz constant and estimation accuracy. Despite all these facts, it still loses to Algorithm 3 both in practice and theory (see Remark 3). Both Algorithms 3 and Algorithm 4 have a single hyperparameter – Lipschitz constant estimation M_p. But since it is more a characteristic of the problem than a particular algorithm hyperparameter, both algorithms should be the same. We conducted several experiments where we compare Algorithm 3 and Algorithm 4 with the same parameter M_p from the set $\{0.01, 0.1, 0.5, 1, 5\}$. The overall picture is the same, and Algorithm 3 outperforms Algorithm 4. Figure 7 shows that Algorithm 3 restarts only in the end, which does not affect the overall result. Since the theoretical estimate for the number of iterations, after which we should restart our algorithm, is usually too pessimistic, we tried to perform restarts manually after every 25 or 50 iterations of the inner algorithm. However, it only worsened the convergence of the whole Algorithm 3.

Finally, algorithm 1 behaves the same way as Algorithm 3, and in the end, it converges even faster. The reason it covers in the restart condition $A_{N_k} \geq \frac{4}{\mu}$: Algorithm 3 chooses to restart when it has almost achieved the solution. Again, one can see it as little spikes at the end of the blue line. Since, in our case, $\mu = \frac{\varepsilon}{4R}$, we can not just directly change the value of μ. We can do this only through ε because R is theoretically estimated and specific to chosen objective problem. However, both the increase and decrease of ε in our experiments showed similar results. Thus, to address this issue, we conduct additional experiments on a strongly convex objective in the next subsection, where we can directly specify the constant of strong convexity μ.

6.4. Minimal Mutual Information

The experiments for entropy regularized optimal transport were not representative of the performance of Algorithm 3 compared to Algorithm 1; in this subsection, we look at the Minimal Mutual Information problem, which is strongly convex. This problem is defined as follows:

$$\min_{x \in \Sigma_n} \left\{ f(x) := \frac{L}{2}\|Ax - b\|_2^2 + \mu \sum_{k=1}^{n} x_k \ln x_k \right\}, \tag{71}$$

where Σ_n is n-dimensional standard simplex.

If we consider this problem in different space $\{(x, z) | z = Ax, x \in \Sigma_n\}$, then it becomes optimization problem with linear equality constraints

$$\min_{\substack{x \in \Sigma_n \\ z = Ax}} \left\{ f(x, z) := \frac{L}{2}\|z - b\|_2^2 + \mu \sum_{k=1}^{n} x_k \ln x_k \right\}. \tag{72}$$

The dual problem to (72) looks as follows

$$\min_{\lambda \in \mathbb{R}^m} \left\{ \varphi(\lambda) := \mu \ln \left(\sum_{i=1}^{n} \exp\left(\frac{\left[-A^T\lambda\right]_i}{\mu} \right) \right) + \frac{1}{2L}\left(\|\lambda + b\|_2^2 - \|b\|_2^2 \right) \right\}. \tag{73}$$

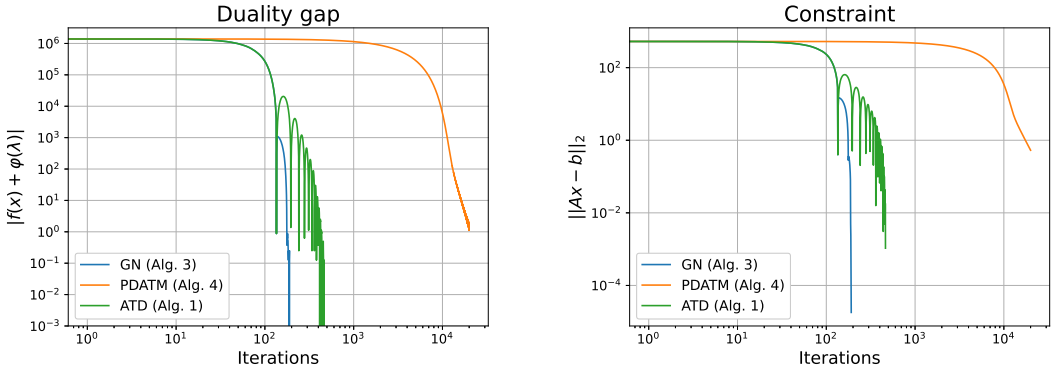
Since, in this case, dual objective (73) is initially strongly convex with constant $\frac{1}{L}$, we do not need additional regularization in Algorithm 3, and we use $f_\mu \equiv \varphi(\lambda),\ \mu = \frac{1}{L}$. We do not provide any additional analysis for the case when the objective of Algorithm 3 is initially strongly convex because the differences with proofs of Theorem 4.2 are minor. The resulting convergence rates can be seen in Remark 1.

In our experiments we used dataset "housing" from [11], scaled to $[-1, 1]$, which we then transfered to $[0, 1]$. We tested several values of M_p. The overall picture was the same, but the convergence of all the algorithms was faster for smaller values of M_p, and the resulting picture was not so evident. Thus, we decided to choose $M_p = 100$. Additionally, we have tested several values of L and μ. Again, the overall picture was the same, but it took too long for some values to converge for all the methods. Thus, we chose $\mu = 1$, $L = 10$. We start from zeros and stop the optimization process when the approximation error achieves $\varepsilon = \varepsilon_f = \varepsilon_{eq} = 0.01$. We derive an estimate of R from strong convexity:

$$\|\lambda^*\|_2 \leq \frac{\|\nabla\varphi(0) - \nabla\varphi(\lambda^*)\|_2}{\mu} = \frac{\|\nabla\varphi(0)\|_2}{\mu} = R.$$

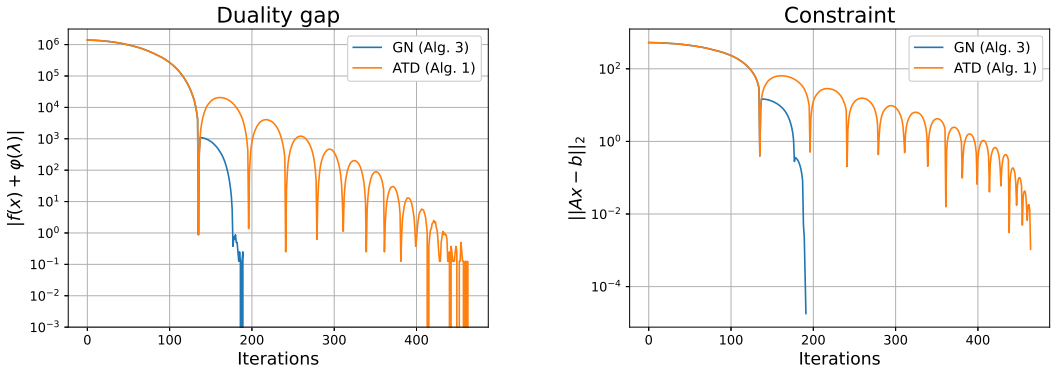
We show the comparison of Algorithms 3, 4 and 1 in Figure 9, and then take a closer look at Algorithms 3 and 1 in Figure 10. Here Algorithm 1 minimizes dual function (73) and recalculates the value of the primal function and equality constraint on every iteration until it reaches approximation error of ε both for duality gap and constraint.

Again, like in MNIST experiments, we can see that Algorithm 3 for gradient norm minimization (GN) outperforms Algorithm 4, that is, primal-dual accelerated tensor method (PDATM). We limit the maximal number of steps for Algorithm 4 to 20000, so



(a) (b)

Figure 9. Duality gap and constraint convergence for Algorithms 3, 4 and 1



(a) (b)

Figure 10. Duality gap and constraint convergence for Algorithms 3 and 1

it stops when it reaches this limit and does not converge to the area of approximation error neither for dual gap (Figure 9a) nor for constraint (Figure 9b). Thus, Algorithm 3 converges more than 100 times faster.

Figure 10 shows that Algorithm 1 (ATD) performs the same way as Algorithm 3 until the first restart. Then they both start jumping: Algorithm 3 due to restarts, and Algorithm 1 – due to the nature of used acceleration. But, our proposed Algorithm 3 converges almost 2.5 times faster. This result shows that our proposed gradient norm minimization framework makes sense when it restarts early.

Remark 5. The above results show that Monteiro-Svaiter acceleration [32], which we used in Algorithm 3, covers all the implementational drawbacks of Algorithm 3, which results in better numerical convergence compared to Algorithm 4. As we mentioned at the end of Remark 4, in Algorithm 4, we use Nesterov acceleration, which gives a worse convergence rate than Monteiro-Svaiter acceleration. Future work should consider a primal-dual method with Monteiro-Svaiter acceleration and compare its performance numerically with Algorithms 3 and 4. This will make the Primal-Dual method not that straightforward because at least it will introduce additional linear search, which comes with Monteiro-Svaiter acceleration.

Remark 6. Comparison of results of Algorithms 3 and 1 on convex entropy-regularized optimal transport (Section 6.3.2) and on strongly convex MMI problem (Section 6.4) showed that it is beneficial to use Algorithm 3, when μ is not too small. Otherwise, the first restart would be done when the method almost achieved the solution or would not be done at all. For example, in Section 6.3.2 we had $\varepsilon = 0.001$, $R \simeq 100 \Rightarrow \mu = \varepsilon/(4R) \sim 10^{-5}$, and in Section 6.4 $\mu = 1/L = 10^{-1}$. That is why the second case method restarts much earlier than it achieves its solution.

7. Conclusions

This paper considers minimization problems with linear equality constraints. There are two ways to solve this type of problem: find a stationary point of dual function and reconstruct the primal solution or use the primal-dual method, which optimizes both primal and dual function simultaneously. We consider both approaches. Firstly, we propose two high-order methods for gradient norm minimization. These methods have optimal convergence rates up to multiplicative logarithmic factors. Secondly, we propose a high-order primal-dual accelerated tensor method that uses Nesterov's acceleration. Finally, we compare these two approaches with each other both in theory and in practice. Additionally, we numerically compare the proposed methods with the primal-dual version of the near-optimal tensor method for convex optimization.

Funding

This work was supported by Ministry of Science and Higher Education grant No. 075-10-2021-068. The work by P. Dvurechensky was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - The Berlin Mathematics Research Center MATH$^+$ (EXC-2046/1, project ID: 390685689). The work of C.A. Uribe is supported in part by the National Science Foundation under Grant No. 2211815 and No. 2213568.

References

[1] N. Agarwal and E. Hazan, *Lower Bounds for Higher-Order Convex Optimization*, in *Proceedings of the 31st Conference On Learning Theory*, S. Bubeck, V. Perchet, and P. Rigollet, eds., Proceedings of Machine Learning Research Vol. 75, 06–09 Jul. PMLR, 2018, pp. 774–792. Available at `http://proceedings.mlr.press/v75/agarwal18a.html`.

[2] A. Alacaoglu, Q. Tran Dinh, O. Fercoq, and V. Cevher, *Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization*, in *Advances in Neural Information Processing Systems 30*, I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Curran Associates, Inc., 2017, pp. 5852–5861.

[3] A.S. Anikin, A.V. Gasnikov, P.E. Dvurechensky, A.I. Tyurin, and A.V. Chernov, *Dual approaches to the minimization of strongly convex functionals with a simple structure under affine constraints*, Computational Mathematics and Mathematical Physics 57 (2017), pp. 1262–1276.

[4] Y. Arjevani, O. Shamir, and R. Shiff, *Oracle complexity of second-order methods for smooth convex optimization*, Mathematical Programming (2018). Available at `https://doi.org/10.1007/s10107-018-1293-1`.

[5] M. Baes, *Estimate sequence methods: extensions and approximations* (2009).

[6] E.G. Birgin, J.L. Gardenghi, J.M. Martínez, S.A. Santos, and P.L. Toint, *Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models*, Mathematical Programming 163 (2017), pp. 359–368. Available at `https://doi.org/10.1007/s10107-016-1065-8`.

[7] 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*. PMLR, 2019, pp. 492–507.

[8] B. Bullins, *Fast minimization of structured convex quartics*, arXiv preprint arXiv:1812.10349 (2018).

[9] B. Bullins and R. Peng, *Higher-order accelerated methods for faster non-smooth optimization*, arXiv preprint arXiv:1906.01621 (2019).

[10] C. Cartis, N.I.M. Gould, and P.L. Toint, *Improved second-order evaluation complexity for unconstrained nonlinear optimization using high-order regularized models*, arXiv:1708.04044 (2018).

[11] C.C. Chang and C.J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology 2 (2011), pp. 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[12] A. Chernov, P. Dvurechensky, and A. Gasnikov, *Fast Primal-Dual Gradient Method for Strongly Convex Minimization Problems with Linear Constraints*, in *Discrete Optimization and Operations Research: 9th International Conference, DOOR 2016, Vladivostok, Russia, September 19-23, 2016, Proceedings*, Y. Kochetov, M. Khachay, V. Beresnev, E. Nurminski, and P. Pardalos, eds. Springer International Publishing, 2016, pp. 391–403.

[13] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, in *Advances in Neural Information Processing Systems 26*, C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, eds., Curran Associates, Inc., 2013, pp. 2292–2300.

[14] M. Cuturi and G. Peyré, *A smoothed dual approach for variational wasserstein problems*, SIAM Journal on Imaging Sciences 9 (2016), pp. 320–343.

[15] J.M. Danskin, *The theory of max-min and its application to weapons allocation problems*, Vol. 5, Springer Science & Business Media, 2012.

[16] D. Dua and C. Graff, *UCI machine learning repository* (2017). Available at `http://archive.ics.uci.edu/ml`.

[17] P. Dvurechensky, A. Gasnikov, E. Gasnikova, S. Matsievsky, A. Rodomanov, and I. Usik, *Primal-Dual Method for Searching Equilibrium in Hierarchical Congestion Population Games*, in *Supplementary Proceedings of the 9th International Conference on Discrete Optimization and Operations Research and Scientific School (DOOR 2016) Vladivostok, Russia, September 19 - 23, 2016*. 2016, pp. 584–595. arXiv:1606.08988.

[18] P. Dvurechensky, A. Gasnikov, and A. Kroshnin, *Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm*, in *International conference on machine learning*. PMLR, 2018, pp. 1367–1376.

[19] A.V. Gasnikov, E.V. Gasnikova, Y.E. Nesterov, and A.V. Chernov, *Efficient numerical methods for entropy-linear programming problems*, Computational Mathematics and Mathematical Physics 56 (2016), pp. 514–524.

[20] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, and C.A. Uribe, *Optimal Tensor Methods in Smooth Convex and Uniformly Convex Optimization*, in *Proceedings of the Thirty-Second Conference on Learning Theory*, A. Beygelzimer and D. Hsu, eds., Proceedings of Machine Learning Research Vol. 99, 25–28 Jun, Phoenix, USA. PMLR, 2019, pp. 1374–1391. Available at `http://proceedings.mlr.press/v99/gasnikov19a.html`, arXiv:1809.00382.

[21] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C.A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, Q. Jiang, Y.T. Lee, Y. Li, and A. Sidford, *Near Optimal Methods for Minimizing Convex Functions with Lipschitz p-th Derivatives*, in *Proceedings of the Thirty-Second Conference on Learning Theory*, A. Beygelzimer and D. Hsu, eds., Proceedings of Machine Learning Research Vol. 99, 25–28 Jun, Phoenix, USA. PMLR, 2019, pp. 1392–1393. Available at

http://proceedings.mlr.press/v99/gasnikov19b.html.

[22] G.N. Grapiglia and Y. Nesterov, *Tensor methods for finding approximate stationary points of convex functions*, Optimization Methods and Software (2020), pp. 1–34.

[23] S.V. Guminov, Y.E. Nesterov, P.E. Dvurechensky, and A.V. Gasnikov, *Accelerated primal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems*, Doklady Mathematics 99 (2019), pp. 125–128. Available at https://doi.org/10.1134/S1064562419020042.

[24] S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov, *On a combination of alternating minimization and Nesterov's momentum*, in *International Conference on Machine Learning*. PMLR, 2021, pp. 3886–3898.

[25] O. Hinder, A. Sidford, and N. Sohoni, *Near-optimal methods for minimizing star-convex functions and beyond*, in *Conference on Learning Theory*. PMLR, 2020, pp. 1894–1938.

[26] K.H. Hoffmann and H.J. Kornstaedt, *Higher-order necessary conditions in abstract mathematical programming*, Journal of Optimization Theory and Applications 26 (1978), pp. 533–568. Available at https://doi.org/10.1007/BF00933151.

[27] D. Kamzolov and A. Gasnikov, *Near-optimal hyperfast second-order method for convex optimization and its sliding*, arXiv preprint arXiv:2002.09050 (2020).

[28] D. Kovalev and A. Gasnikov, *The first optimal acceleration of high-order methods in smooth convex optimization*, arXiv preprint arXiv:2205.09647 (2022).

[29] T. Lin, N. Ho, X. Chen, M. Cuturi, and M.I. Jordan, *Computational Hardness and Fast Algorithm for Fixed-Support Wasserstein Barycenter*, arXiv:2002.04783 (2020).

[30] T. Lin, N. Ho, M. Cuturi, and M.I. Jordan, *On the Complexity of Approximating Multimarginal Optimal Transport*, arXiv:1910.00152 (2019).

[31] T. Lin, N. Ho, and M. Jordan, *On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms*, in *International Conference on Machine Learning*. PMLR, 2019, pp. 3982–3991.

[32] R. Monteiro and B. Svaiter, *An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods*, SIAM Journal on Optimization 23 (2013), pp. 1092–1125. Available at https://doi.org/10.1137/110833786.

[33] Y. Nesterov, *Accelerating the cubic regularization of newton's method on convex problems*, Mathematical Programming 112 (2008), pp. 159–181. Available at https://doi.org/10.1007/s10107-006-0089-x.

[34] Y. Nesterov, *Introductory Lectures on Convex Optimization: a basic course*, Kluwer Academic Publishers, Massachusetts, 2004.

[35] Y. Nesterov, *Smooth minimization of non-smooth functions*, Mathematical Programming 103 (2005), pp. 127–152.

[36] Y. Nesterov, *How to make the gradients small*, Optima. Mathematical Optimization Society Newsletter (2012), pp. 10–11.

[37] Y. Nesterov, *Implementable tensor methods in unconstrained convex optimization*, Mathematical Programming 186 (2021), pp. 157–183.

[38] Y. Nesterov, *Inexact accelerated high-order proximal-point methods*, Mathematical Programming (2021), pp. 1–26.

[39] Y. Nesterov, *Inexact high-order proximal-point methods with auxiliary search procedure*, SIAM Journal on Optimization 31 (2021), pp. 2807–2828.

[40] Y. Nesterov, *Superfast second-order methods for unconstrained convex optimization*, Journal of Optimization Theory and Applications 191 (2021), pp. 1–30.

[41] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky, *Primal-dual accelerated gradient methods with small-dimensional relaxation oracle*, Optimization Methods and Software (2020), pp. 1–28.

[42] C. Song and Y. Ma, *Towards unified acceleration of high-order algorithms under hölder continuity and uniform convexity*, arXiv:1906.00582 (2019).

[43] Q. Tran-Dinh, A. Alacaoglu, O. Fercoq, and V. Cevher, *An adaptive primal-dual framework for nonsmooth convex minimization*, Mathematical Programming Computation 12 (2020), pp. 451–491.

[44] Q. Tran-Dinh and V. Cevher, *Constrained Convex Minimization via Model-based Excessive Gap*, in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA. MIT Press, NIPS'14, 2014, pp. 721–729.

[45] Q. Tran-Dinh, O. Fercoq, and V. Cevher, *A smooth primal-dual optimization framework for nonsmooth composite convex minimization*, SIAM Journal on Optimization 28 (2018), pp. 96–134. Available at `https://doi.org/10.1137/16M1093094`, arXiv:1507.06243.

[46] A. Wibisono, A.C. Wilson, and M.I. Jordan, *A variational perspective on accelerated methods in optimization*, Proceedings of the National Academy of Sciences 113 (2016), pp. E7351–E7358.

[47] A. Wilson, L. Mackey, and A. Wibisono, *Accelerating rescaled gradient descent*, arXiv preprint arXiv:1902.08825 (2019).

[48] A. Yurtsever, Q. Tran-Dinh, and V. Cevher, *A Universal Primal-dual Convex Optimization Framework*, in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA. MIT Press, NIPS'15, 2015, pp. 3150–3158.