Gradient Methods for Problems with Inexact Model of the Objective

Fedor S. Stonyakin[1,4(✉)], Darina Dvinskikh[2,3], Pavel Dvurechensky[2,3], Alexey Kroshnin[3,4], Olesya Kuznetsova[4], Artem Agafonov[4], Alexander Gasnikov[3,4,5], Alexander Tyurin[5], César A. Uribe[6], Dmitry Pasechnyuk[7], and Sergei Artamonov[5]

[1] V.I. Vernadsky Crimean Federal University, Simferopol, Russia
fedyor@mail.ru
[2] Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany
{darina.dvinskikh,pavel.dvurechensky}@wias-berlin.de
[3] Institute for Information Transmission Problems RAS, Moscow, Russia
gasnikov@yandex.ru
[4] Moscow Institute of Physics and Technologies, Moscow, Russia
{kroshnin,kuznetsova.oa,agafonov.ad}@phystech.edu
[5] National Research University Higher School of Economics, Moscow, Russia
alexandertiurin@gmail.com, sartamonov@hse.ru
[6] Massachusetts Institute of Technology, Cambridge, USA
cauribe@mit.edu
[7] 239-th School of St. Petersburg, Saint Petersburg, Russia
pasechnyuk2004@gmail.com

Abstract. We consider optimization methods for convex minimization problems under inexact information on the objective function. We introduce inexact model of the objective, which as a particular cases includes inexact oracle [16] and relative smoothness condition [36]. We analyze gradient method which uses this inexact model and obtain convergence rates for convex and strongly convex problems. To show potential applications of our general framework we consider three particular problems. The first one is clustering by electorial model introduced in [41]. The second one is approximating optimal transport distance, for which we propose a Proximal Sinkhorn algorithm. The third one is devoted to approximating optimal transport barycenter and we propose a Proximal Iterative Bregman Projections algorithm. We also illustrate the practical performance of our algorithms by numerical experiments.

Keywords: Gradient method · Inexact oracle · Strong convexity · Relative smoothness · Bregman divergence

1 Introduction

In this paper we consider optimization methods for convex problems under inexact information on the objective function. This information is given by an object,

M. Khachay et al. (Eds.): MOTOR 2019, LNCS 11548, pp. 97–114, 2019.
https://doi.org/10.1007/978-3-030-22629-9_8

which we call *inexact model*. Inexact model generalizes the inexact oracle introduced in [16], where inexactness is assumed to be present in the objective value and its gradient. The authors show that, based on these two objects, it is possible to construct a linear function, which is a lower approximation and, up to a quadratic term, an upper approximation of the objective, and these two approximations are enough to obtain convergence rates for gradient method and accelerated gradient method. We go beyond and assume that the approximations of the objective are given through some function, which is not necessarily linear.

This allows us to construct general gradient-type method which is applicable in for different problem classes and allows to obtain convergence rates in these situations as a corollary of our general theorem. Besides convex problems we focus also on strongly convex objectives and illustrate the application of our general theory by two examples. The first example is data clustering by electoral model [41]. The second example relates to Wasserstein distance and barycenter, which are widely used in data analysis [12,13].

Many optimization methods use some model of the objective function to define a step by minimization of this model. Usually the model is constructed using exact first-order [18,39,43], second-order [42], or higher-order information [9,40] information on the objective. The influence of inexactness on the convergence of gradient-type methods have being studied at least since [46]. Accelerated first-order methods with inexact oracle are studied in [11,14,16,21,37]. Some recent works study also non-convex problems in this context [8,19]. Randomized methods with inexact oracle are also studied in the literature, e.g. coordinate descent in [27,53], random gradient-free methods and random directional derivative methods in [22,23]. A method with inexact oracle for variational inequalities can be found in [26].

The contributions of this paper can be summarized as follows.

- We introduce an inexact model of the objective function for convex optimization problems and strongly convex optimization problems.
- We introduce and theoretically analyze a gradient-type method for convex and strongly convex problems with an inexact model of the objective function. For the latter case we prove linear rate of convergence.
- We apply our method to, generally speaking, non-convex optimization problem which arises in clustering model introduced in [41]. To do this we construct an inexact model and apply our general algorithms and convergence theorems.
- We apply our general framework for Wasserstein distance and barycenter problems and show that it allows to construct a proximal á la [10] version of the Sinkhorn's algorithm [49] and Iterative Bregman Projection algorithm [5].

Notation. We define $\mathbf{1} = (1, ..., 1)^T \in \mathbb{R}^n$, $KL(z|t)$ to be the Kullback-Leibler divergence: $KL(z|t) = \sum\limits_{k=1}^{n} z_k \ln(z_k/t_k)$, $\forall z, t \in S_n(1)$, where $S_n(1)$ is the standard simplex in $\mathbb{R}^n$. We also denote by $\odot$ the entrywise product of two matrices.

2 Gradient Methods with Inexact Model of the Objective

Consider the convex optimization problem

$$f(x) \to \min_{x \in Q}, \tag{1}$$

where function f is convex and $Q \subseteq \mathbb{R}^n$ is a simple convex compact set. Moreover, assume that $\min_{x \in Q} f(x) = f(x_*)$ for some $x_* \in Q$.

To solve this problem, we introduce a norm $\|\cdot\|$ on $\mathbb{R}^n$ and a prox-function $d(x)$ which is continuous and convex. We underline that, unlike most of the literature, we do not require d to be strongly convex.

Without loss of generality, we assume that $\min\limits_{x \in \mathbb{R}^n} d(x) = 0$. Further, we define *Bregman divergence* $V[y](x) := d(x) - d(y) - \langle \nabla d(y), x - y \rangle$. Next we define the inexact model of the objective function, which generalizes the inexact oracle of [16] (see also [8,21,24,29,52,54]).

Definition 1. *Let function $\psi_\delta(x, y)$ be convex in $x \in Q$ and satisfy $\psi_\delta(x, x) = 0$ for all $x \in Q$.*

(i) We say that $\psi_\delta(x, y)$ is a (δ, L)-model of the function f at a given point y with respect to $V[y](x)$ iff, for all $x \in Q$, the inequality

$$0 \le f(x) - (f(y) + \psi_\delta(x, y)) \le LV[y](x) + \delta$$

holds for some $L, \delta > 0$.

(ii) We say that $\psi_\delta(x, y)$ is a (δ, L, μ)-model of the function f at a given point y with respect to $V[y](x)$ iff, for all $x \in Q$, the inequality

$$\mu V[y](x) \le f(x) - (f(y) + \psi_\delta(x, y)) \le LV[y](x) + \delta \tag{2}$$

Note that we allow L to depend on δ. We refer to the case (i) as convex case and to the case (ii) as strongly convex case.

Remark 1. In the particular case of function f possessing (δ, L)-oracle [16] at a given point y, one has

$$0 \le f(x) - f(y) - \langle g_\delta(y), x - y \rangle \le \frac{L}{2} \|x - y\|^2 + \delta$$

and $\psi_\delta(x, y) = \langle g_\delta(y), x - y \rangle$. In the same way, if function f is equipped with (δ, L, μ)-oracle [17], i.e.,

$$\frac{\mu}{2} \|x - y\|^2 \le f(x) - f(y) - \langle g_{\delta,L,\mu}(y), x - y \rangle \le \frac{L}{2} \|x - y\|^2 + \delta \quad \forall x \in Q,$$

we have $\psi_\delta(x, y) = \langle g_{\delta,L,\mu}(y), x - y \rangle$.

The algorithms we develop are based on solving auxiliary simple problems on each iteration. We assume that these problems can be solved inexactly and, following [4] introduce a definition of inexact solution of a problem.

Definition 2. *Consider a convex minimization problem*

$$\phi(x) \to \min_{x \in Q \subseteq \mathbb{R}^n}. \tag{3}$$

If ϕ is smooth, we say that we solve it with $\widetilde{\delta}$-'precision' ($\widetilde{\delta} \geq 0$) if we find $\tilde{x}$ s.t. $\max_{x \in Q} \langle \nabla \phi(\tilde{x}), \tilde{x} - x \rangle = \widetilde{\delta}$. If ϕ is general convex, we say that we solve this problem with $\widetilde{\delta}$-'precision' if we find $\tilde{x}$ s.t. $\exists h \in \partial \phi(\widetilde{x})$, $\langle h, x_* - \widetilde{x} \rangle \geq -\widetilde{\delta}$. In both cases we denote this $\tilde{x}$ as $\operatorname{argmin}^{\widetilde{\delta}}_{x \in Q} \phi(x)$.

We notice that the case $\widetilde{\delta} = 0$ corresponds to the case when $\tilde{x}$ is an exact solution of convex optimization problem (3) [4,39].

The connection of Definition 2 with standard definitions of inexact solution, e.g. in terms of the objective residual, can be found in Appendix G of the full version of the paper [51].

2.1 Convex Case

In this subsection we describe a gradient-type method for problems with (δ, L)-model of the objective. This algorithm is a natural extension of gradient method, see [29,52,54].

Algorithm 1. Gradient method with (δ, L)-model of the objective.

1: **Input:** x_0 is the starting point, $L > 0$ and $\delta, \widetilde{\delta} > 0$.
2: **for** $k \geq 0$ **do**
3:
$$\phi_{k+1}(x) := \psi_\delta(x, x_k) + LV[x_k](x), \quad x_{k+1} := \arg\min^{\widetilde{\delta}}_{x \in Q} \phi_{k+1}(x).$$
4: **end for**
Output: $\bar{x}_N = \frac{1}{N} \sum_{k=0}^{N-1} x_{k+1}$

Theorem 1. *Let $V[x_0](x_*) \leq R^2$, where x_0 is the starting point, and x_* is the nearest minimum point to the point x_0 in the sense of Bregman divergence $V[y](x)$. Then, for the sequence, generated by Algorithm 1 the following inequality holds:*

$$f(\bar{x}_N) - f(x_*) \leq \frac{LR^2}{N} + \widetilde{\delta} + \delta,$$

In Appendix A of the full version of the paper [51] we prove this theorem and provide an adaptive version of Algorithm 1, which does not require knowledge of the constant L.

2.2 Strongly Convex Case

In this subsection we consider problem (1) with (δ, L, μ)-model of the objective function satisfying (2). This more strong assumption allows us to obtain linear rate of convergence of the proposed algorithm. Our algorithm is listed as Algorithm 2 and it is a version of Algorithm 1, which is adaptive to possibly unknown constant L.

Algorithm 2. Adaptive gradient method with an oracle using the (δ, L, μ)-model

1: **Input:** x_0 is the starting point, $\mu > 0$ $L_0 \geq 2\mu$ and δ.
2: Set $S_0 := 0$
3: **for** $k \geq 0$ **do**
4: Find the smallest $i_k \geq 0$ such that

$$f(x_{k+1}) \leq f(x_k) + \psi_\delta(x_{k+1}, x_k) + L_{k+1} V[x_k](x_{k+1}) + \delta,$$

where $L_{k+1} = 2^{i_k - 1} L_k$ for $L_k \geq 2\mu$ and $L_{k+1} = 2^{i_k} L_k$ for $L_k < 2\mu$, $\alpha_{k+1} := \frac{1}{L_{k+1}}$, $S_{k+1} := S_k + \alpha_{k+1}$.

$$\phi_{k+1}(x) := \psi_\delta(x, x_k) + L_{k+1} V[x_k](x), \quad x_{k+1} := \arg\min_{x \in Q}^{\widetilde{\delta}} \phi_{k+1}(x).$$

5: **end for**
Output: $\bar{x}_N = \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{x_{k+1}}{L_{k+1}}$

Let's introduce average parameter $\hat{L}$:

$$1 - \frac{\mu}{\hat{L}} = \sqrt[k+1]{\left(1 - \frac{\mu}{L_{k+1}}\right)\left(1 - \frac{\mu}{L_k}\right) \ldots \left(1 - \frac{\mu}{L_1}\right)}.$$

Note that by $L_i \geq \mu$ $(i = 1, 2, \ldots)$

$$\min_{1 \leq i \leq k+1} L_i \leq \hat{L} \leq \max_{1 \leq i \leq k+1} L_i \leq 2L.$$

The following result holds.

Theorem 2. *Let $\psi_\delta(x, y)$ is a (δ, L, μ)-model for f w.r.t. $V[y](x)$. Then, after k iterations of Algorithm 2, we have*

$$V[x_{k+1}](x_*) \leq \frac{2L(\delta + \widetilde{\delta})}{\mu^2}\left(1 - \left(1 - \frac{\mu}{2L}\right)^{k+1}\right) + \left(1 - \frac{\mu}{\hat{L}}\right)^{k+1} V[x_0](x_*),$$

$$f(x_{k+1}) - f(x_*) \leq \frac{4L^2(\delta + \widetilde{\delta})}{\mu^2}\left(1 - \left(1 - \frac{\mu}{2L}\right)^{k+1}\right) + 2L\left(1 - \frac{\mu}{\hat{L}}\right)^{k+1} V[x_0](x_*).$$

The details of proof can be found in Appendix B of the full version of the paper [51]. Note that Algorithm 1 also has linear convergence rate for the strongly

convex case. The details can be found in Appendix C of the full version of the paper [51]. The benefit of Algorithm 1 is that there is no need to know the strong convexity parameter μ for the algorithm to work. On the other hand, this parameter is needed for assessing the quality of the solution returned by the algorithm. The benefit of the adaptive version is that it does not require to know the value of the parameter L and adapts to it. Moreover, the parameter L can be different for the model at different points and the algorithm adapts also for the local value of this parameter.

3 Clustering by Electorial Model

In this section we consider clustering model introduced in [41]. In this model voters (data points) choose a party (cluster) in an iterative manner by alternative minimization of the following function.

$$f_{\mu_1,\mu_2}(x=(z,p)) = g(x) + \mu_1 \sum_{k=1}^{n} z_k \ln z_k + \frac{\mu_2}{2}\|p\|_2^2 \to \min_{z\in S_n(1), p\in\mathbb{R}^m_+}, \quad (4)$$

where $\mathbb{R}^m_+$ is a non-negative orthant and $S_n(1)$ is the standard n-dimensional simplex in $\mathbb{R}^n$.

The vector z contains probabilities with which voters choose the considered party, and vector p describes the position of the party in the space of voter opinions. The minimized potential is the result of combining two optimization problems into one: voters choose the party whose position is closest to their personal opinion and the party adjusts its position minimizing dispersion and trying not to go too far from its initial position. Yu. Nesterov in [41] used sequential elections process to show that under some natural assumptions the process convergence and gives the clustering of the data-points. This was done for a particular choice of the function g which has limited interpretability. We show, how our framework of inexact model of the objective allows to construct a gradient-type method for the case of general function g, which is not necessarily convex.

Assume that $g(x)$ (generally, non-convex) is a function with L_g-Lipschitz continuous gradient:

$$\|\nabla g(x) - \nabla g(y)\|_* \le L_g\|x-y\| \quad \forall x, y \in S_n(1)\times\mathbb{R}^m_+,$$

and, following [41], the numbers μ_1, μ_2 are chosen such that $L_g \le \mu_1$ and $L_g \le \mu_2$.

The norm $\|\cdot\|$ in $S_n(1)\times\mathbb{R}^m_+$ is defined as $\|(z,p)\|^2 = \|z\|_1^2 + \|p\|_2^2$, where $\|z\|_1 = \sum\limits_{k=1}^{n} z_k$ and $\|p\|_2 = \sqrt{\sum\limits_{k=1}^{m} p_k^2}$. The correctness of this definition is proven in Appendix I of the full version of the paper [51].

It can be shown that

$$\psi_\delta(x,y) = \langle \nabla g(y), x-y\rangle - L_g \cdot KL(z_x|z_y) - \frac{L_g}{2}\|p_x - p_y\|_2^2 + \mu_1(KL(z_x|\mathbf{1}) - KL(z_y|\mathbf{1})) + \frac{\mu_2}{2}\left(\|p_x\|_2^2 - \|p_y\|_2^2\right)$$

is a $(0, 2L_g)$-model of $f_{\mu_1,\mu_2}(x)$ in x with respect to the following Bregman divergence

$$V[y](x) = KL(z_x|z_y) + \frac{1}{2}\|p_x - p_y\|_2^2.$$

The proof is detailed in Appendix I of the full version of the paper [51].

Further, for the case $\min\{\mu_1,\mu_2\} > L_g$ $\psi_\delta(x,y)$ is a strongly convex w.r.t. $V[y](x)$:

$$\psi_\delta(x,y) = \psi_\delta^{lin}(x,y) + (\mu_1 - L_g)\cdot KL(z_x|z_y) + \frac{\mu_2 - L_g}{2}\|p_x - p_y\|_2^2 \geq (\min\{\mu_1,\mu_2\} - L_g)\cdot V[y](x), \tag{5}$$

where

$$\psi_\delta^{lin}(x,y) = \langle \nabla g(y), x-y\rangle + \mu_1\langle \nabla KL(z_y|1), z_x - z_y\rangle + \mu_2\langle p_y, p_x - p_y\rangle$$

is linear in y. The proof of (5) is given in Appendix I of the full version of the paper [51].

Thus, $\psi_\delta^{lin}(x,y)$ is a $(0, \max\{\mu_1,\mu_2\} + L_g, \min\{\mu_1,\mu_2\} - L_g)$-model of the function f_{μ_1,μ_2}:

$$f_{\mu_1,\mu_2}(y) + \psi_\delta^{lin}(x,y) + (\min\{\mu_1,\mu_2\} - L_g)V[y](x) \leq f_{\mu_1,\mu_2}(x)$$

and

$$f_{\mu_1,\mu_2}(x) \leq f_{\mu_1,\mu_2}(y) + \psi_\delta^{lin}(x,y) + (\max\{\mu_1,\mu_2\} + L_g)V[y](x).$$

So, we can apply our Algorithms 1 and 2 to the problem (4).

4 Proximal Sinkhorn Algorithm for Optimal Transport

In this section we consider the problem of approximating an optimal transport (OT) distance. Recently optimal transport distances has gained a lot of interest in machine learning and statistical applications [3,6,15,28,34,45,50]. To state the OT problem, assume that we are given two discrete probability measures $p, q \in S_n(1)$ and ground cost matrix $C \in \mathbb{R}_+^{n\times n}$, then the optimal transport problem is

$$\langle C, \pi\rangle \to \min_{\pi\in\mathcal{U}(p,q)}, \quad \mathcal{U}(p,q) = \{\pi \in \mathbb{R}_+^{n\times n} : \pi\mathbf{1} = p, \pi^T\mathbf{1} = q\} \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes Frobenius dot product of matrices, π is a transportation plan.

The above optimal transport problem is the Kantorovich [31] linear program (LP) formulation of the problem, which goes back to the Monge's problem [38].

The best known theoretical complexity for this linear program is[1] $\widetilde{O}(n^{2.5})$, see [35]. However, there is no known practical implementation of this algorithm. In practice, the simplex method gives complexity $O(n^3 \ln n)$ [44]. We follow the alternative approach based on entropic regularization of the OT problem [12]. We show how our general framework of inexact model of the objective allows to construct Proximal Sinkhorn algorithm with better computational stability in comparison with the standard Sinkhorn algorithm.

For any optimization problem (1), $\psi_\delta(x, y) = f(x) - f(y)$ satisfies Definition 1 with any $L \geq 0$. In this case, our Algorithm 1 becomes inexact *Bregman proximal gradient method*

$$x_{k+1} = \arg\min_{x \in Q}^{\tilde{\delta}} \{f(x) + LV[x_k](x)\}.$$

Our idea is to apply this proximal method for the OT problem and approximately find the next iterate x_{k+1} by Sinkhorn's algorithm [2,12,25,49]. The latter is made possible by the choice of V as KL divergence, which makes the problem of finding the point x_{k+1} to be an entropy-regularized OT problem, which, in turn, is efficiently solvable by the Sinkhorn algorithm.

Consider the iterates

$$\pi^0 = pq^T \in \mathcal{U}(p,q), \quad \pi^{k+1} = \arg\min_{\pi \in \mathcal{U}(p,q)}^{\varepsilon/2} \left\{\langle C, \pi \rangle + L \cdot KL(\pi | \pi^k)\right\}$$
$$= \arg\min_{\pi \in \mathcal{U}(p,q)}^{\varepsilon/2} KL\left(\pi \middle| \pi^k \odot \exp\left(-\frac{C}{L}\right)\right), \quad (7)$$

which we call outer iterations. On each outer iteration we use Sinkhorn's algorithm 3, which solves the minimization problem in (7) with accuracy $\tilde{\varepsilon}$ in terms of its objective residual. Notice that here ε' differs from the one from [2,25] as we need approximated solution to the regularized problem. Moreover, unlike [25] we use a slightly refined theoretical bounds for the Sinkhorn's algorithm not depending on vectors p, q[2].

Theorem 3. *Let $\bar{\pi}^N = \frac{1}{N}\sum_{k=1}^N \pi^k$, where π^k are the iterates of (7). Then, after $N = \frac{4L \ln n}{\varepsilon}$ iterations, it holds that $\langle C, \bar{\pi}^N \rangle \leq \min_{\pi \in \mathcal{U}(p,q)} \langle C, \pi \rangle + \varepsilon$. Moreover, the accuracy $\tilde{\varepsilon}$ for the solution of (7) is sufficient to be set as $\widetilde{O}(\varepsilon^4/(Ln^4))$ and the complexity of Sinkhorn's Algorithm on k-th iteration is bounded as*

$$n^2 \widetilde{O}\left(\min\left\{\exp\left(\frac{\bar{c}_k}{L}\right)\left(\frac{\bar{c}_k}{L} + \ln\frac{\bar{c}_k}{\tilde{\varepsilon}}\right), \frac{\bar{c}_k^2}{L\tilde{\varepsilon}}\right\}\right), \quad (8)$$

[1] Here and below for all (large) n: $\widetilde{O}(g(n)) \leq \tilde{C} \cdot (\ln n)^r g(n)$ with some constants $\tilde{C} > 0$ and $r \geq 0$. Typically, $r = 1$, but not in this particular case. If $r = 0$, then $\widetilde{O}(\cdot) = O(\cdot)$.

[2] One can find the proof in Appendix E of the full version of the paper [51].

Algorithm 3. Sinkhorn's Algorithm

Input: Accuracy $\tilde{\varepsilon}$, matrix $K = e^{-C/\gamma}$, marginals $p, q \in S_n(1)$.

Set $t = 0$, $u^0 = \ln p$, $v^0 = \ln q$, $\varepsilon' = \frac{\tilde{\varepsilon}}{4}\left(\max_{i,j} C_{ij} - \min_{i,j} C_{ij} + 2\gamma \ln\left(\frac{4\gamma n^2}{\tilde{\varepsilon}}\right)\right)^{-1}$.
repeat
  if $t \bmod 2 = 0$ **then**
    $u^{t+1} = u^t + \ln p - \ln(B(u^t, v^t)\mathbb{1})$, where $B(u, v) := \operatorname{diag}(e^u) K \operatorname{diag}(e^v)$
    $v^{t+1} = v^t$
  else
    $v^{t+1} = v^t + \ln q - \ln(B(u^t, v^t)^T \mathbb{1})$
    $u^{t+1} = u^t$
  end if
  $t = t + 1$
until $\left\|B(u^t, v^t)\mathbb{1} - p\right\|_1 + \left\|B(u^t, v^t)^T\mathbb{1} - q\right\|_1 \leq \varepsilon'$
Find $\hat{\pi}$ as the projection of $B(u^t, v^t)$ on $\mathcal{U}(p, q)$ by Algorithm 2 in [2].

Output: $\hat{\pi}$.

where[3]

$$\bar{c}_k = \|C\|_\infty + L \ln\left(\frac{\max_{i,j} \pi^k_{ij}}{\min_{i,j} \pi^k_{ij}}\right).$$

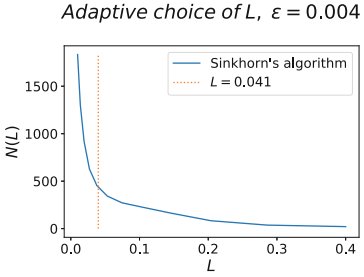

Fig. 1. Adaptive choice of L

Remark 2. The standard Sinkhorn's method can be seen as a particular case of our algorithm (7) with only one step. To obtain an ε-approximate solution of (6), the regularization parameter L needs to be chosen $O(\varepsilon / \ln n)$ [2,25,30]. This can lead to instability of the Sinkhorn's algorithm [48]. On the opposite, our Proximal Sinkhorn algorithm allows to run Sinkhorn's algorithm with larger regularization parameter. This parameter can be chosen by minimization of the theoretical bound (8), which gives $L = \widetilde{O}(\|C\|_\infty)$. In practice one can choose this constant adaptively since we have a (δ, L)-model for any L and can vary L from iteration to iteration. First, the inner problem (7) is solved with overestimated L. Then, we set $L := L/2$ and the problem is solved with the updated value of the parameter and so on until a significant increase (e.g. 10 times) in the complexity of the auxiliary entropy-linear programming problem in comparison with the initial complexity

[3] This bound is rough and typically $\bar{c}_k$ is smaller in practice. By proper rounding of π^k one can guarantee (without loss of generality) that $\pi^k_{ij} \geq \varepsilon/(2n^2 \|C\|_\infty)$, which gives

$$\frac{\bar{c}_k}{L} = \frac{\|C\|_\infty}{L} + \ln\left(\frac{2n^2 \|C\|_\infty}{\varepsilon}\right).$$

But, in practice there often is no need to make 'rounding' after each outer iteration.

is detected, see Fig. 1, where $N(L)$ is a number of required iterations of Sinkhorn algorithm to solve the inner problem with accuracy ε.

From the Theorem 3 and Remark 2 one can roughly estimate the total complexity of Proximal Sinkhorn algorithm as[4] $\widetilde{O}(n^4/\varepsilon^2)$.

We also mention several recent complexity bounds[5] for the OT problem $\tilde{O}(n^2/\varepsilon^3)$ [2], $\tilde{O}(n^2/\varepsilon^2)$ and $\tilde{O}(n^{2.5}/\varepsilon)$ [25], $\tilde{O}(n^2/\varepsilon)$ [7,47], $\tilde{O}(n/\varepsilon^{3+d})$, $d \geq 1$ [1].

4.1 Numerical Illustration

In this subsection we provide numerical illustration of the Proximal Sinkhorn algorithm.[6] In the experiments we use a standard MNIST dataset with images scaled to a size 10×10. The vectors p and q contain the pixel intensities of the first and second images respectively. The value of c_{ij} is equal to the Euclidean distance between the i-th pixel from the vector p and the j-th pixel from the vector q on the image pixel grid. For experiments with varying number of pixels n the images are resized to be images of $10 \cdot m \times 10 \cdot m$ pixels, where $m \in \mathbb{N}$. We replace all the zero elements in p and q with 10^{-3} and, then, normalize these vectors.

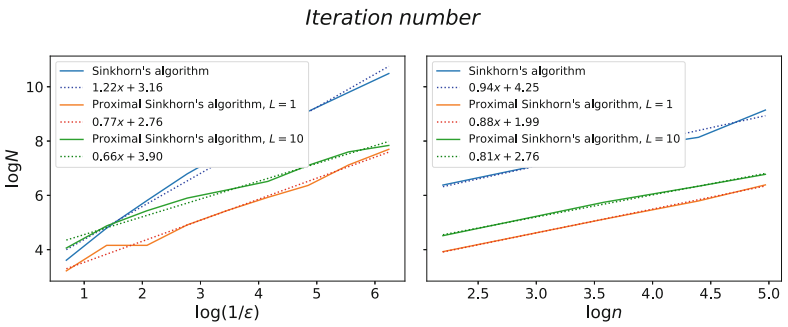

Fig. 2. Comparison of iteration number of Sinkhorn's algorithm and total number of Sinkhorn steps in Proximal Sinkhorn's algorithm for different L.

Figure 2 shows that the growth rate of the iteration number with increasing accuracy or size of the problem for the Sinkhorn's algorithm is greater than for

[4] Our experiments on MNIST data set show (see Figs. 2, 3) that in practice the bound is better.

[5] Strictly speaking for the moment we can not verify all the details of the proof of estimate $\tilde{O}(n^2/\varepsilon)$. Also the proposed in [7,47] methods are mainly theoretical, like Lee–Sidford's method for OT problem with the complexity $\tilde{O}(n^{2.5})$ [35]. For the moment it is hardly possible to implement these methods such that theirs practical efficiencies correspond to the theoretical ones.

[6] The code is available at https://github.com/dmivilensky/Proximal-Sinkhorn-algorithm.

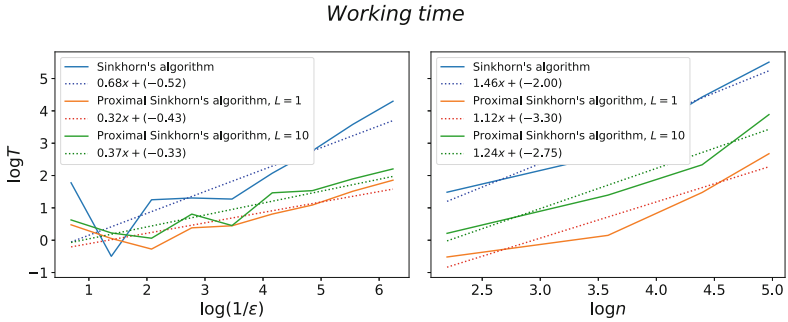

Fig. 3. Comparison of working time of Sinkhorn's algorithm and Proximal Sinkhorn's algorithm with different L.

the Proximal Sinkhorn's method. At the same time, with a higher value of L in proximal method, the iteration number is greater, and the growth rates with some precision are equal. The same type of dependence on the accuracy and the size of the problem can be seen for the working time (Fig. 3).

More experiments can be found in the full version of this paper [51], in particular, on the mean number of inner iterations[7].

5 Proximal IBP Algorithm for Wasserstein Barycenter

In this section we consider a more complicated problem of approximating an OT barycenter. OT barycenter is a natural definition of a mean in a space endowed with an OT distance. Such barycenters are used in the analysis of data with geometric structure, e.g. images, and other machine learning applications [5,13,32,33,45].

For a set of probability measures $\{p_1, \ldots, p_m\}$, cost matrices $C_1, \ldots, C_m \in \mathbb{R}_+^{n \times n}$, and $w \in S_n(1)$, the weighted barycenter of these measures is defined as a solution of the following convex optimization problem

$$\sum_{l=1}^{m} w_l \min_{\pi_l \in \mathcal{U}(p_l, q)} \langle C_l, \pi_l \rangle \to \min_{q \in S_n(1)} \iff \sum_{l=1}^{m} w_l \langle C_l, \pi_l \rangle \to \min_{\boldsymbol{\pi} \in \mathcal{C}_1 \cap \mathcal{C}_2},$$

$$\mathcal{C}_1 = \{\boldsymbol{\pi} = [\pi_1, \ldots, \pi_m] : \forall l \ \pi_l \mathbb{1} = p_l\}, \quad \mathcal{C}_2 = \left\{\boldsymbol{\pi} = [\pi_1, \ldots, \pi_m] : \pi_1^T \mathbb{1} = \cdots = \pi_m^T \mathbb{1}\right\}.$$

The idea is similar to the one in Sect. 4, namely, we use our framework to define a Proximal Iterative Bregman Projections algorithm.

[7] Figures 5–8 are given in the more complete version of the text by link https://arxiv.org/abs/1902.09001

The algorithm starts from the point $\boldsymbol{\pi}$ s.t. $\pi_l^0 = \frac{1}{n} p_l \mathbb{1}^T \in \mathcal{U}(p_l, \mathbb{1}/n)$, $l = 1, ..., m$ and iterates

$$\boldsymbol{\pi}^{k+1} = \arg\min_{\boldsymbol{\pi} \in \mathcal{C}_1 \cap \mathcal{C}_2}{}^{\varepsilon/2} \sum_{l=1}^{m} w_l \left\{ \langle C_l, \pi_l \rangle + L \cdot KL(\pi_l | \pi_l^k) \right\}$$
$$= \arg\min_{\boldsymbol{\pi} \in \mathcal{C}_1 \cap \mathcal{C}_2}{}^{\varepsilon/2} \sum_{l=1}^{m} w_l KL\left(\pi_l \left| \pi_l^k \odot \exp\left(-\frac{C_l}{L} \right) \right. \right). \tag{9}$$

These iterations are called outer iterations and on each such iteration, the Iterative Bregman Projections algorithm [5] listed as Algorithm 4 below is used to solve the auxiliary minimization problem.

Algorithm 4. Iterative Bregman Projection

Input: $C_1, \ldots, C_m$, $p_1, \ldots, p_m$, $L > 0$, $\tilde{\varepsilon} > 0$

1: $u_l^0 := 0$, $v_l^0 := 0$, $K_l := \exp\left(-\frac{C_l}{L}\right)$, $l = 1, \ldots, m$
2: **repeat**
3: $v_l^{t+1} := \sum_{k=1}^m w_k \ln K_k^T e^{u_k^t} - \ln K_l^T e^{u_l^t}$, $\mathtt{u}^{t+1} := \mathtt{u}^t$
4: $t := t + 1$
5: $u_l^{t+1} := \ln p_l - \ln K_l e^{v_l^t}$, $\mathtt{v}^{t+1} := \mathtt{v}^t$
6: $t := t + 1$
7: **until** $\sum_{l=1}^m w_l \left\| B_l^T(u_l^t, v_l^t)\mathbb{1} - \bar{q}^t \right\|_1 \leq \frac{\tilde{\varepsilon}}{4 \max_l \|C_l\|_\infty}$, where $B_l(u_l, v_l) = \operatorname{diag}(e^{u_l}) K_l \operatorname{diag}(e^{v_l})$, $\bar{q}^t := \sum_{l=1}^m w_l B_l^T(u_l^t, v_l^t)\mathbb{1}$
8: $q := \frac{1}{\sum_{l=1}^m w_l \langle \mathbb{1}, B_l \mathbb{1} \rangle} \sum_{l=1}^m w_l B_l^T \mathbb{1}$
9: Calculate $\hat{\pi}_1, \ldots, \hat{\pi}_m$ by Algorithm 2 from [2] s.t. $\hat{\pi}_l \in \mathcal{U}(p_l, q)$, $\|\hat{\pi}_l - B_l\|_1 \leq \|B_l \mathbb{1} - p_l\|_1 + \left\|B_l^T \mathbb{1} - q\right\|_1$.

Output: q, $\hat{\boldsymbol{\pi}} = [\hat{\pi}_1, \ldots, \hat{\pi}_m]$.

Theorem 4. *Let $\bar{\boldsymbol{\pi}}^N = \frac{1}{N} \sum_{k=1}^N \boldsymbol{\pi}^k$, where $\boldsymbol{\pi}^k$ are the iterates of (9). Then, after $N = \frac{4Lm \ln n}{\varepsilon}$ iterations, it holds that*

$$\sum_{l=1}^{m} w_l \langle C_l, \bar{\pi}_l^N \rangle \leq \min_{\boldsymbol{\pi} \in \mathcal{C}_1 \cap \mathcal{C}_2} \sum_{l=1}^{m} w_l \langle C_l, \pi_l \rangle + \varepsilon.$$

Moreover, the accuracy $\tilde{\varepsilon}$ for the solution of (9) is sufficient to be set as $\tilde{\varepsilon} = \widetilde{O}(\varepsilon^2/(mn^3))$ and the complexity of IBP on k-th iteration is bounded as

$$mn^2 \widetilde{O}\left(\min\left\{ \exp\left(\frac{\bar{c}_k}{L}\right) \ln \frac{\bar{c}_k}{\tilde{\varepsilon}}, \frac{\bar{c}_k^2}{L\tilde{\varepsilon}} \right\} \right),$$
$$\bar{c}_k = O\left(\max_{l=1,\ldots,m} \left[\|C_l\|_\infty + L \ln\left(\frac{\max_{i,j}[\pi_l^k]_{ij}}{\min_{i,j}[\pi_l^k]_{ij}} \right) \right] \right).$$

The proof of Theorem 4 is based on Theorem 1 and [32]. All the remarks from Sect. 4 for Proximal Sinkhorn algorithm also hold for Proximal IBP.

In [32] it was shown that complexity of IBP is $\widetilde{O}\left(n^2/\varepsilon^2\right)$. Despite the theoretical complexity of Proximal IBP is worse than this bound, we show in the next section that in practice Proximal IBP beats the standard IBP algorithm. As an alternative to the IBP algorithm we mention primal-dual accelerated gradient descent [20,55].

5.1 Numerical Illustration

In this section, we present preliminary computational results for the numerical performance analysis of the Proximal Iterative Bregman Projection (ProxIBP) method discussed above asthe iterates (9).

Initially, we show the results for the computation of a non-regularized Wasserstein barycenter of a set of 10 truncated Gaussian distributions with finite support. For the finite support $x = [-5, -4.9, -4.8, \ldots, -0.1, 0, 0.1, \ldots, 4.8, 4.9, 5]$, we set the finite distribution p_l such that $p_l(i) = \mathcal{N}(x_i; \mu_i, \sigma_i)$, that is, the value at coordinate i of the distribution p_l, for $1 \leq l \leq m$, is the value of the Normal distribution with mean μ_i and standard deviation σ_i. The values $\{\mu_i\} \sim \text{Uniform}[-5, 5]$, are uniformly chosen in the line segment $[-5, 5]$, and the values are selected as $\{\mu_i\} \sim \text{Uniform}[0.25, 1.25]$. For simplicity of exposition, we select uniform weighting for all distributions, i.e., $w_l = 1/m$.

Figure 4 shows the numerical results for a number of comparative scenarios between the Iterative Bregman Projection (IBP) algorithm proposed in [5] and its Proximal variant in (9). For both algorithms, we show the function values achieved by the generated iterates, and the final approximated barycenter. The results for the IBP algorithm are shown in Fig. 4(a) and (b). Figure 4(a) shows the weighted distance between the generated barycenter and the original distributions for three different desired accuracy values.

It is clear that a bigger ε generates a faster convergence, but the final cost is slightly higher than in other cases. Figure 4(b) shows the resulting barycenter for the three values of the accuracy parameter. For higher accuracy, the effects of the regularization constant are smaller and thus we obtain a "spikier" barycenter. Figure 4(c) and (d) shows a similar analysis for the proposed Proximal IBP in (9), in Fig. 4(c) we observe the function value of the generated barycenter, for a fixed number of inner loop iterations, and changing values of L, note that here L is not a regularization parameter but the weight on the Bregman function. For larger values of L, the inner loop problem is easier to solve, requires less iterations to achieve certain accuracy, with the price in a larger number of iterations in the outer loop. For the particular problem studied, 200 iterations in the outer loop are sufficient to achieve good performance even with relatively smaller values of L. Figure 4(c) shows the generated barycenters for the Proximal IBP algorithm. Finally, Fig. 4(e) and (f) show the results, for the analogous adaptive stopping condition described in Line 11 of Algorithm 3 with $\varepsilon = 1 \cdot 10^{-10}$. We test two different values of the parameter L, namely 1 and 0.1. Additionally, we explore the suggested adaptive search procedure, where one decreases the value of the parameter L at each iteration, until the inner problem has become particularly hard to solve. This last approach is shown a fast convergence as

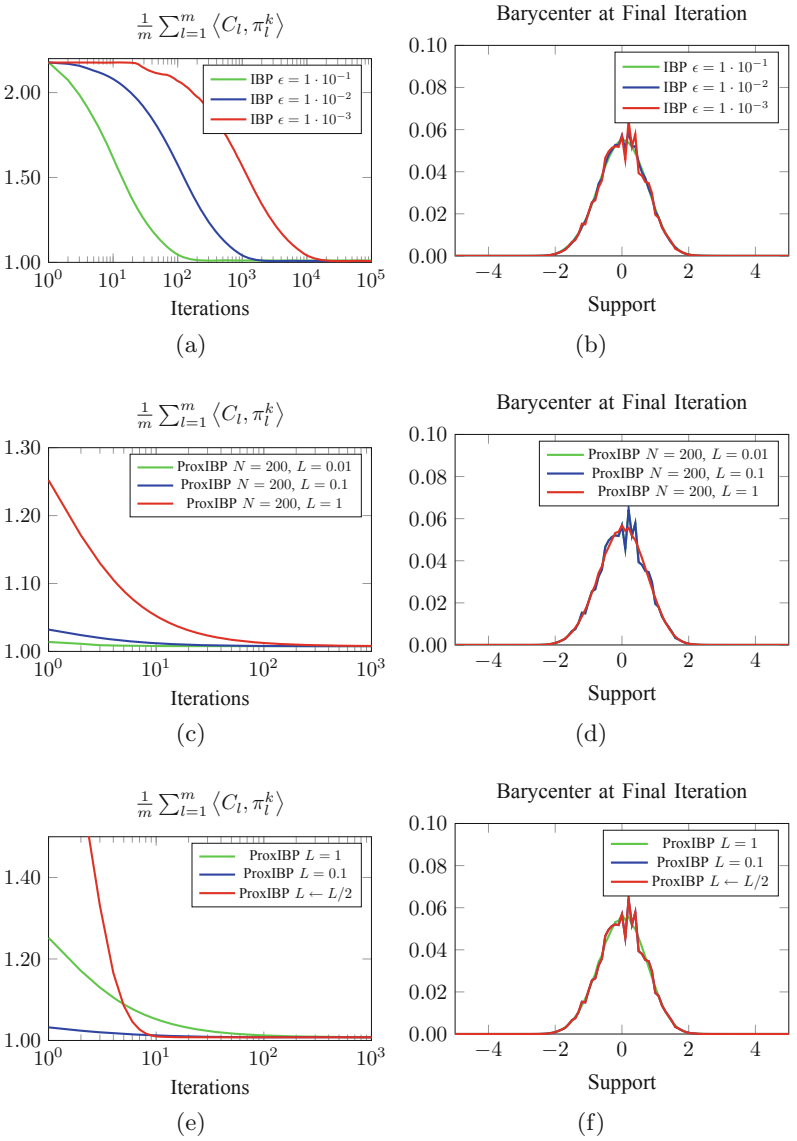

Fig. 4. Numerical results for the computation of the barycenter of 10 truncated Gaussian random variables with finite support for the IBP Algorithm and the Proximal IBP algorithm. Both function value and final resulting barycenter are shown for an number of simulation scenarios.

it reaches a comparable value in around 10 iterations. Figure 4(f) shows the resulting barycenters.

Again, we refer to the full version [51] for additional experiments e.g. on computing Wasserstein barycenters of images from MNIST dataset[8].

6 Conclusions

In this paper we consider gradient methods with inexact information of the objective given by inexact model of this objective. We analyze a gradient-type method for this type of problems and provide its convergence rate. To illustrate the applications, we consider optimization problems in optimal transport and a clustering model. Notably, our framework allows to solve non-convex problems which have a convex inexact model, which is illustrated in the section devoted to clustering model.

Acknowledgments. The work in Sects. 4 and 5 was funded by Russian Science Foundation (project 18-71-10108). The work in Subsect. 2.1 and Sect. 3 was supported by Russian Foundation for Basic Research 18-31-20005 mol_a_ved. The work of F. Stonyakin on Algorithm 2 and Theorem 2 was supported by Russian Science Foundation (project 18-71-00048). The work of A. Gasnikov in Sect. 2 was supported within the framework of the HSE University Basic Research Program and funded by the Russian Academic Excellence Project "5-100". The work of A. Kroshnin in Sect. 3 was supported within the framework of the HSE University Basic Research Program and funded by the Russian Academic Excellence Project "5-100". The work of S. Artamonov in Sect. 3 was supported by Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2019–2020 (grant No 19-01-024) and by the Russian Academic Excellence Project "5-100".

References

1. Altschuler, J., Bach, F., Rudi, A., Weed, J.: Approximating the quadratic transportation metric in near-linear time. arXiv:1810.10046 (2018)
2. Altschuler, J., Weed, J., Rigollet, P.: Near-linear time approxfimation algorithms for optimal transport via sinkhorn iteration. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 1961–1971. Curran Associates, Inc. (2017)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv:1701.07875 (2017)
4. Ben-Tal, A., Nemirovski, A.: Lectures on modern convex optimization (lecture notes). Personal web-page of A. Nemirovski (2015). http://www2.isye.gatech.edu/~nemirovs/Lect_ModConvOpt.pdf
5. Benamou, J.D., Carlier, G., Cuturi, M., Nenna, L., Peyré, G.: Iterative bregman projections for regularized transportation problems. SIAM J. Sci. Comput. **37**(2), A1111–A1138 (2015)
6. Bigot, J., Klein, T., et al.: Consistent estimation of a population barycenter in the Wasserstein space. arXiv:1212.2562 (2012)

[8] Figures 5–8 are given in the more complete version of the text by link https://arxiv.org/abs/1902.09001

7. Blanchet, J., Jambulapati, A., Kent, C., Sidford, A.: Towards optimal running times for optimal transport. arXiv:1810.07717 (2018)
8. Bogolubsky, L., et al.: Learning supervised PageRank with gradient-based and gradient-free optimization methods. In: NIPS 2016 (2016). http://papers.nips.cc/paper/6565-learning-supervised-pagerank-with-gradient-based-and-gradient-free-optimization-methods.pdf
9. Cartis, C., Gould, N.I.M., Toint, P.L.: Improved second-order evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. arXiv:1708.04044 (2018)
10. Chen, G., Teboulle, M.: Convergence analysis of a proximal-like minimization algorithm using bregman functions. SIAM J. Optim. **3**(3), 538–543 (1993)
11. Cohen, M.B., Diakonikolas, J., Orecchia, L.: On acceleration with noise-corrupted gradients. arXiv:1805.12591 (2018)
12. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 2292–2300. Curran Associates, Inc. (2013)
13. Cuturi, M., Doucet, A.: Fast computation of Wasserstein barycenters. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, Bejing, China, 22–24 June 2014, pp. 685–693. PMLR (2014). http://proceedings.mlr.press/v32/cuturi14.html
14. d'Aspremont, A.: Smooth optimization with approximate gradient. SIAM J. Optim. **19**(3), 1171–1183 (2008). https://doi.org/10.1137/060676386
15. Del Barrio, E., Lescornel, H., Loubes, J.M.: A statistical analysis of a deformation model with Wasserstein barycenters: estimation procedure and goodness of fit test. arXiv:1508.06465 (2015)
16. Devolder, O., Glineur, F., Nesterov, Y.: First-order methods of smooth convex optimization with inexact oracle. Math. Program. **146**(1), 37–75 (2014). https://doi.org/10.1007/s10107-013-0677-5
17. Devolder, O., Glineur, F., Nesterov, Y., et al.: First-order methods with inexact oracle: the strongly convex case. CORE Discussion Papers **2013016** (2013)
18. Drusvyatskiy, D., Ioffe, A.D., Lewis, A.S.: Nonsmooth optimization using taylor-like models: error bounds, convergence, and termination criteria. arXiv:1610.03446 (2016)
19. Dvurechensky, P.: Gradient method with inexact oracle for composite non-convex optimization. arXiv:1703.09180 (2017)
20. Dvurechensky, P., Dvinskikh, D., Gasnikov, A., Uribe, C.A., Nedić, A.: Decentralize and randomize: faster algorithm for Wasserstein barycenters. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 10783–10793. NeurIPS 2018, Curran Associates, Inc. (2018). arXiv:1802.04367
21. Dvurechensky, P., Gasnikov, A.: Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. J. Optim. Theory Appl. **171**(1), 121–145 (2016). https://doi.org/10.1007/s10957-016-0999-6
22. Dvurechensky, P., Gasnikov, A., Gorbunov, E.: An accelerated directional derivative method for smooth stochastic convex optimization. arXiv:1804.02394 (2018)
23. Dvurechensky, P., Gasnikov, A., Gorbunov, E.: An accelerated method for derivative-free smooth stochastic convex optimization. arXiv:1802.09022 (2018)
24. Dvurechensky, P., Gasnikov, A., Kamzolov, D.: Universal intermediate gradient method for convex problems with inexact oracle. arXiv:1712.06036 (2017)

25. Dvurechensky, P., Gasnikov, A., Kroshnin, A.: Computational optimal transport: complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1367–1376 (2018). arXiv:1802.04367
26. Dvurechensky, P., Gasnikov, A., Stonyakin, F., Titov, A.: Generalized Mirror Prox: Solving variational inequalities with monotone operator, inexact oracle, and unknown Hölder parameters (2018). https://arxiv.org/abs/1806.05140
27. Dvurechensky, P., Gasnikov, A., Tiurin, A.: Randomized similar triangles method: a unifying framework for accelerated randomized optimization methods (coordinate descent, directional search, derivative-free method) (2017). https://arxiv.org/abs/1707.08486
28. Ebert, J., Spokoiny, V., Suvorikova, A.: Construction of non-asymptotic confidence sets in 2-Wasserstein space (2017). https://arxiv.org/abs/1703.03658
29. Gasnikov, A.: Universal gradient descent (2017). https://arxiv.org/abs/1711.00394
30. Gasnikov, A., et al.: Universal method with inexact oracle and its applications for searching equillibriums in multistage transport problems (2015). https://arxiv.org/abs/1506.00292
31. Kantorovich, L.: On the translocation of masses. Doklady Acad. Sci. USSR (N.S.) **37**(7–8), 227–229 (1942)
32. Kroshnin, A., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., Tupitsa, N., Uribe, C.: On the complexity of approximating Wasserstein barycenter (2019). https://arxiv.org/abs/1901.08686
33. Kroshnin, A., Spokoiny, V., Suvorikova, A.: Statistical inference for bures-Wasserstein barycenters (2019). https://arxiv.org/abs/1901.00226
34. Le Gouic, T., Loubes, J.M.: Existence and consistency of Wasserstein barycenters. Probab. Theory Relat. Fields **168**(3–4), 901–917 (2017)
35. Lee, Y.T., Sidford, A.: Path finding methods for linear programming: solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science Foundations of Computer Science (FOCS), pp. 424–433 (2014)
36. Lu, H., Freund, R.M., Nesterov, Y.: Relatively smooth convex optimization by first-order methods, and applications. SIAM J. Optim. **28**(1), 333–354 (2018)
37. Mairal, J.: Optimization with first-order surrogate functions. In: International Conference on Machine Learning, pp. 783–791 (2013)
38. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris (1781)
39. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, Massachusetts (2004)
40. Nesterov, Y.: Implementable tensor methods in unconstrained convex optimization. CORE Discussion Papers 2018005, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), March 2018. https://ideas.repec.org/p/cor/louvco/2018005.html
41. Nesterov, Y.: Soft clustering by convex electoral model. CORE Discussion Papers 2018001, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), January 2018. https://ideas.repec.org/p/cor/louvco/2018001.html
42. Nesterov, Y., Polyak, B.: Cubic regularization of Newton method and its global performance. Math. Program. **108**(1), 177–205 (2006)
43. Ochs, P., Fadili, J., Brox, T.: Non-smooth non-convex bregman minimization: unification and new algorithms. J. Optim. Theory Appl. **181**(1), 244–278 (2019)

44. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467 (2009)
45. Peyré, G., Cuturi, M.: Computational optimal transport. Found. Trends Mach. Learn. **11**(5–6), 355–607 (2019)
46. Polyak, B.: Introduction to Optimization. Optimization Software, New York (1987)
47. Quanrud, K.: Approximating optimal transport with linear programs. In: 2nd Symposium on Simplicity in Algorithms (SOSA 2019), vol. 69, pp. 6:1–6:9. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018)
48. Schmitzer, B.: Stabilized Sparse Scaling Algorithms for Entropy Regularized Transport Problems (2016). https://arxiv.org/abs/1610.06519
49. Sinkhorn, R.: Diagonal equivalence to matrices with prescribed row and column sums. II. Proc. Amer. Math. Soc. **45**(2), 195–198 (1974)
50. Solomon, J., Rustamov, R.M., Guibas, L., Butscher, A.: wasserstein propagation for semi-supervised learning. In: Proceedings of the 31st International Conference on International Conference on Machine Learning, vol. 32, pp. 306–314. PMLR (2014)
51. Stonyakin, F., et al.: Gradient methods for problems with inexact model of the objective. arXiv:1902.09001 (2019)
52. Stonyakin, F., et al.: Inexact Model: A Framework for Optimization and Variational Inequalities (2019). https://arxiv.org/abs/1902.00990
53. Tappenden, R., Richtárik, P., Gondzio, J.: Inexact coordinate descent: complexity and preconditioning. J. Optim. Theory Appl. **170**(1), 144–176 (2016)
54. Tyurin, A., Gasnikov, A.: Fast gradient descent method for convex optimization problems with an oracle that generates a (δ, L)-model of a function in a requested point. Comput. Math. Math. Phys. (2019, accepted). https://arxiv.org/abs/1711.02747
55. Uribe, C.A., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., Nedić, A.: Distributed computation of Wasserstein barycenters over networks. In: 2018 IEEE Conference on Decision and Control (CDC), pp. 6544–6549 (2018)