Oracle Complexity Separation in Strongly Convex-Concave Saddle Point Optimization

Chris Junchi Li[◇]

Department of Electrical Engineering and Computer Sciences[◇]
University of California, Berkeley

September 19, 2024

Abstract

In this work, we present novel algorithms for solving strongly convex-concave saddle point problems (SPPs) with a composite structure, where both the primal and dual functions exhibit smoothness and strong convexity-concavity properties. The problem is framed as minimizing a composite function involving separable terms for the primal and dual variables, while the coupling term is bilinear. Our primary contribution is an algorithm that achieves the lower bounds for iteration complexity while optimally separating the oracle complexities between the composite terms and the saddle part. We demonstrate the practical implications of these results in distributed optimization, personalized federated learning, and empirical risk minimization. The proposed algorithm significantly reduces communication rounds and oracle calls, providing an efficient solution for large-scale optimization problems. We also validate the performance through numerical experiments, illustrating the superiority of our approach.

Keywords: Convex-Concave Saddle Point Problems; Minimax Optimization; Separable Oracle Complexity; Iteration Complexity; Distributed Optimization

1 Introduction

Convex-concave saddle point problems (SPPs) arise in a variety of domains, including machine learning, optimization, game theory, and distributed computing. These problems can be structured as minimax formulations where one function is minimized over one set of variables and maximized over another. This class of problems, particularly when the objective functions exhibit strong convexity in one direction and strong concavity in the other, presents unique challenges due to its duality and composite structure. An efficient solution for these problems is crucial in real-world applications where scalability and optimal performance are essential.

In this work, we focus on solving strongly convex-concave SPPs that involve composite terms, making the problem more complex by coupling smooth primal and dual functions with non-proximal-friendly terms. Such structures appear in various practical contexts, including distributed optimization and federated learning, where decentralized computations must occur under communication constraints. Furthermore, composite minimization problems, such as empirical risk minimization in machine learning, benefit from saddle point formulations, enabling more efficient algorithmic solutions.

We aim to address the gap in separating the oracle complexities of the composite terms and the saddle part. Specifically, we propose an efficient sliding algorithm that leverages this structure to achieve optimal iteration and oracle complexity. Our method provides a significant improvement over traditional approaches that do not exploit the separability of the oracle complexities, reducing computational overhead in applications that require large-scale optimization.

Mathematically, we consider strongly convex and strongly concave SPPs with the composite structure:

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} p(x) + R(x,y) - q(y) \tag{1}$$

where $p(x) : \mathbb{R}^{d_x} \to \mathbb{R}$, $q(y) : \mathbb{R}^{d_y} \to \mathbb{R}$ are convex and L_p, L_q-smooth function respectively and $R(x,y) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ is L_R-smooth, μ_x-strongly convex and μ_y-strongly concave. Both composites $p(x)$, $q(y)$ are not necessarily proximal friendly. Note, that we can also consider $R(x,y)$ to be convex-concave and $p(x)$ is μ_x-strongly convex and $q(y)$ is μ_y-strongly concave. By the transformation $p(x) \to p(x) - \frac{\mu_x}{2}\|x\|^2$, $q(y) \to q(y) + \frac{\mu_y}{2}\|y\|^2$ and $R(x,y) \to R(x,y) + \frac{\mu_x}{2}\|x\|^2 - \frac{\mu_y}{2}\|y\|^2$ we can reduce this case to the problem (1).

The lower bounds of iteration complexity for the problem (1) $\Omega\left(\left(\sqrt{\frac{L_p}{\mu_x}} + \frac{L_R}{\sqrt{\mu_x \mu_y}} + \sqrt{\frac{L_q}{\mu_y}}\right) \log \frac{1}{\varepsilon}\right)$ was proposed in [ZHZ19]. In this work we present algorithm that achieve these lower bounds. But the focus of this work is on the composites complexity separation which is a key issue in many applications. Below, we give some prime examples of this.

Distributed optimization. One of the classic application of the problem (1) is a decentralized distributed optimization over communication network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$\min_{W\mathbf{x}=0} \sum_{i=1}^{n} f_i(x_i) \tag{2}$$

where $\mathbf{x} = [x_1^\top, \ldots, x_n^\top]^\top$, $n = |\mathcal{V}|$ is the number of nodes (clients) in $\mathcal{G}$, $\{f_i(x_i)\}_{i=1}^n$ are functions that store on nodes with variables x_i. Also client i can communicate with client j if and only if there is edge in graph $\mathcal{G}$, i.e. $(i,j) \in \mathcal{E}$ and W is a gossip matrix for communication network $\mathcal{G}$ which responsible for communications between nodes. In particular, the Laplacian matrix of $\mathcal{G}$ can be used as W. Note, that to solve problem in this formulation we have to use conditional optimization methods. To move to unconditional optimization we use a penalty function $\psi(x)$:

$$\min_{\mathbf{x} \in \mathbb{R}^{nd}} F(\mathbf{x}) + \psi(W\mathbf{x}) \tag{3}$$

where $F(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$ is μ_F-strongly convex, L_F-smooth function and $\psi(\mathbf{y}) = 0$ if $\mathbf{y} = 0$, in otherwise $\psi(\mathbf{y}) = +\infty$. In many practical examples, solving the dual problem to (3) preferably. This problem has the following form:

$$\min_{\mathbf{x} \in \mathbb{R}^{nd}} \max_{\mathbf{y} \in \mathbb{R}^{nd}} F(\mathbf{x}) + \langle \mathbf{y}, W\mathbf{x} \rangle \tag{4}$$

due to $\psi(\mathbf{y})$ is indicator function and $\psi^*(\mathbf{y}) = 0$. Meanwhile, to solve problem (4) we need to find gradient $\nabla F(x)$, i.e. compute local gradients $\nabla f_i(x_i)$. Also, we need to compute $W\mathbf{x}$, $W\mathbf{y}$ (gradients of saddle part $\langle \mathbf{y}, W\mathbf{x} \rangle$), i.e. make the communication round. The usually goal for this problem is reduce the communication rounds [SNRar], [BMR+20]. It means, that separation of the oracle complexities to composite and saddle part in problem (4) is significant problem.

Personalized federated learning. The other important example of the problem (1) is personalized federated saddle point problem [SCST17, WWKS18, LFYL20, GDG19]

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} \frac{\lambda}{2} \left\|\sqrt{W}X\right\|^2 + \sum_{m=1}^{M} f_m(x_m, y_m) - \frac{\lambda}{2} \left\|\sqrt{W}Y\right\|^2 \tag{5}$$

where $x_1, \ldots, x_M$ and $y_1, \ldots, y_M$ are interpreted as local models on nodes which are grouped into matrices $X := [x_1, \ldots, x_M]^\top$ and $Y := [y_1, \ldots, y_M]^\top$. $\lambda > 0$ is the key regularization parameter, which corresponds to the personalising degree of the models and W is the gossip matrix reflecting the properties of the communication graph between the nodes. As mentioned above, the composite gradient oracles WX, WY are responsible for the communications. Since, we are interested in separating and reducing gradient calls of composites.

Empirical risk minimization. The other important practical case of the problem (1) is Emperical Risk Minimization problem. This example comes from machine learning [SSBDar]. This problem has the following form

$$\min_{x \in \mathbb{R}^{d_x}} p(x) + q(Bx) \tag{6}$$

where $q(x)$ is convex loss function, B is matrix with data features and $p(x)$ is strongly convex regularizer. This problem is equal to the following saddle point problem

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} p(x) + x^\top By - q^*(y) \tag{7}$$

which can be preferable than problem (6) in many practical applications. For example, in distributed optimization to reduce the communication complexity [XYLC19]. Moreover, the gradients $\nabla p(x)$ or $\nabla q^*(y)$ can be difficult to calculate. In this case, we are interested in separating oracle calls.

These practical examples illustrate the importance of separating oracle complexities which lead to the following research question for the problem (1) ***Can we effectively separate oracle complexities for composites and saddle parts?***

1.1 Contributions

Our high-level contributions are summarized as follows: (i) We develop a novel sliding algorithm for solving strongly convex-concave saddle point problems with composite terms. The proposed algorithm achieves lower bounds for iteration complexity and optimally separates the oracle calls for composite functions and the saddle part. (ii) We apply the algorithm to several key examples, including distributed optimization, personalized federated learning, and empirical risk minimization, demonstrating significant communication round reductions. (iii) We validate our theoretical findings through extensive numerical experiments, which show the practical benefits of the proposed approach in real-world datasets.

Motivated by this research question we introduce the novel algorithm that achieves the lower bound on iteration complexity and optimally separate oracle calls for composite and saddle part. Below we provide more detailed contributions of this work.

(i) **New method.** We develop a novel algorithm (Algorithm 1) used a new sliding idea. We present the idea and convergence analysis of Algorithm 1. Moreover, Algorithm 1 has optimal iteration complexity and optimal complexity separation.

(ii) **Best rates.** To the best of our knowledge, we are the first who present optimal algorithm for composite saddle point problem with complexity separation for the non-symmetric case (i.e., $\mu_x \neq \mu_y$).

(iii) **Bilinear case.** We adopt our approach to bilinear saddle point problem. Our Algorithm achieve the lower bounds for iteration and oracle complexities (up to logarithmic factor). Moreover, we formulate these results for distributed optimization.

In the special case, when $R(x, y) = x^\top By$, (1) has been also widely studied, dating at least to the classic work of [CP11] (imaging inverse problems). Modern applications can be find in decentralized optimization [RYKG22, CRG23]. Quadratic variant of the problem (1) also appeared in reinforcement learning [DCL$^+$17].

1.2 Organization

The remainder of the paper is organized as follows: Section 2 provides preliminaries and notation used throughout the paper. In Section 3, we introduce our proposed algorithm and present the main theoretical results. Finally, Section 5 concludes the paper with future directions and remarks.

2 Preliminaries

In this section, we introduce some notation and necessary assumptions used throughout the paper.

- L_p, L_q, L_R - smoothness constants of functions $p(x), q(y)$ and $R(x, y)$ respectively
- $L = \max\{L_p, L_q, L_R\}$
- μ_x - strong convexity constant of $R(x, y)$ for fixed y
- μ_y - strong concavity constant of $R(x, y)$ for fixed x
- $\mathcal{D}_x, \mathcal{D}_y$ - constants such that $\|x^*\| \leq \mathcal{D}_x$, $\|y^*\| \leq \mathcal{D}_y$, $\mathcal{D} = \max\{\mathcal{D}_x, \mathcal{D}_y\}$
- μ_p, μ_q - strong convexity and strong concavity constants of $p(x), q(y)$ respectively.

  (For *bilinear case*)
- $\lambda_{\max}(BB^\top) > 0$ and $\lambda_{\min}(BB^\top) \geq 0$ are maximum and minimum eigenvalue of $BB^\top$ respectively
- $L = \max(L_p, L_q, \sqrt{\lambda_{\max}(BB^\top)})$

Notation. We denote by $\|\cdot\|$ the standard Euclidean norms. We say that a function f is L-smooth on $\mathbb{R}^d$ if its gradient is Lipshitz-continuous, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \tag{8}$$

for some $L > 0$ and any $x, y \in \mathbb{R}^d$. We say that a function f is μ-strongly convex on $\mathbb{R}^d$ if, for some $\mu > 0$ and any $x, y \in \mathbb{R}^d$ it holds that

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|x - y\|^2 \tag{9}$$

We say that a pair $(\widehat{x}, \widehat{y})$ is an ε-solution to (1) if $\|\widehat{x} - x^*\|^2 + \|\widehat{y} - y^*\|^2 \leq \varepsilon$ where (x^*, y^*) is solution to (1). Also, we define the iteration complexity of an algorithm for solving problem (1) as the number of iterations the algorithm requires to find an ε-solution of this problem. To present the idea of Algorithm 1 we introduce $\mathrm{prox}_{\eta f}(\widehat{x})$ operator

$$\mathrm{prox}_{\eta f}(\widehat{x}) = \arg\min_x \left\{ f(x) + \frac{1}{2\eta}\|x - \widehat{x}\|^2 \right\} \tag{10}$$

Meanwhile, we say that a function $f(x)$ is proximal friendly, if we can compute $\mathrm{prox}_{\eta f}(\widehat{x})$ (solve problem (10)) for any point $\widehat{x}$ explicitly or in $\mathcal{O}(1)$ computer calculations.

Last, we need the following notation. For scalar multiplication with non-Euclidean matrix we use $\langle x, y\rangle_A := x^\top A y$. Also, we use the following matrix:

$$P := \begin{pmatrix} \frac{1}{\eta_x} I & 0 \\ 0 & \frac{1}{\eta_y} I \end{pmatrix}, \quad P^{-1} := \begin{pmatrix} \eta_x I & 0 \\ 0 & \eta_y I \end{pmatrix}$$

Finally, we state the assumptions that we impose on problem (1).

Assumption 1. *$p(x) : \mathbb{R}^{d_x} \to \mathbb{R}$ is L_p-smooth and convex on $\mathbb{R}^{d_x}$.*

Assumption 2. *$q(y) : \mathbb{R}^{d_y} \to \mathbb{R}$ is L_q-smooth and convex on $\mathbb{R}^{d_y}$.*

Assumption 3. *$R(x, y) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ is L_R-smooth on $\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, μ_x-strongly convex on $\mathbb{R}^{d_x}$ for fixed y and μ_y-strongly concave on $\mathbb{R}^{d_y}$ for fixed x.*

3 Optimal Algorithm

In this section, we present the key idea to develop Algorithm 1. After that we provide iteration and oracle complexities of Algorithm 1. Finally, we propose these complexities to the problem (1) with one composite.

3.1 Idea

To understand the idea of Algorithm 1 we temporarily switch from the composite saddle point problem (1) to the composite minimization problem:

$$\min_x u(x) + v(x) \tag{11}$$

with μ-strongly convex function $u(x) + v(x)$ and L_u, L_v-smooth functions $u(x), v(x)$ respectively. The basic and natural way to solve this problem is apply Nesterov's Accelerated Gradient Descent [Nesar], [Nes18] in the following form from [Nes18]:

$$\begin{aligned} x_g^k &= \alpha x^k + (1-\alpha) x_f^k \\ x^{k+1} &= x^k - \eta(\nabla u(x_g^k) + \nabla v(x_g^k)) \\ x_f^{k+1} &= x_g^k - \widetilde{\beta}(\nabla u(x_g^k) + \nabla v(x_g^k)) \end{aligned} \tag{12}$$

This method can be rewrite in equivalent form:

$$\begin{aligned} x_g^k &= \alpha x^k + (1-\alpha) x_f^k \\ x^{k+1} &= x^k - \eta(\nabla u(x_g^k) + \nabla v(x_g^k)) \\ x_f^{k+1} &= x_g^k + \beta(x^{k+1} - x^k) \end{aligned} \tag{13}$$

with $\beta = \frac{\widetilde{\beta}}{\eta}$. The oracle complexity for this method is $\mathcal{O}\left(\sqrt{\frac{L_u+L_v}{\mu}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla u(x)$ and $\nabla v(x)$ to find an ε-solution to (11). This approach does not allow to separate the oracles' complexities that, as mentioned above, may be important in some cases. For example, if we additionally assume that $v(x)$ is proximal friendly function the lower bounds for problem (11) are $\Omega\left(\sqrt{\frac{L_u}{\mu}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla u(x)$. Consequently, for this case Nesterov's Accelerated Gradient Descent is not optimal method since it does not use the effectiveness computation of $\text{prox}_{\eta v}(\cdot)$. Due to we can apply more optimal method Accelerated Proximal Point Algorithm [Roc76], [Gul91], [AT06], [Tse08], [LW16]:

$$\begin{aligned}
x_g^k &= \alpha x^k + (1-\alpha)x_f^k \\
x^{k+1} &= \text{prox}_{\eta v}(x^k - \eta\nabla u(x_g^k)) \\
x_f^{k+1} &= x_g^k + \beta(x^{k+1} - x^k)
\end{aligned}$$

which based on Nesterov's Accelerated Gradient Descent in form (13). This method required $\mathcal{O}\left(\sqrt{\frac{L_u}{\mu}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla u(x)$ and $\text{prox}_{\eta v}(\cdot)$ to find an ε-solution to the problem (11). Moreover, Accelerated Proximal Point Algorithm does not require smoothness of function $v(x)$ and can be applied to problem (11) with non-smooth composites. To adapt this approach for L_v-smooth and non-proximal friendly function $v(x)$ we rewrite $x^{k+1} = \text{prox}_{\eta v}(x^k - \eta\nabla u(x_g^k))$ for diffirentiable function $v(x)$ in implicit form:

$$x^{k+1} = -\eta\nabla v(x^{k+1}) + x^k - \eta\nabla u(y^k) \tag{14}$$

using the first optimal condition. Meanwhile, for non-proximal friendly function we can compute $\text{prox}_{\eta v}(x^k - \eta\nabla u(y^k))$ approximately with Accelerated Gradient Descent and compute x^{k+1} by (14) using this solution. Summing up the above, we get the following method which based on sliding technique [Lan16]

$$\begin{aligned}
x_g^k &= \alpha x^k + (1-\alpha)x_f^k \\
\widehat{x}^{k+1} &\approx \text{prox}_{\eta v}(x^k - \eta\nabla u(x_g^k)) \\
x^{k+1} &= x^k - \eta(\nabla u(x_g^k) + \nabla v(\widehat{x}^{k+1})) \\
x_f^{k+1} &= x_g^k + \beta(\widehat{x}^{k+1} - x^k)
\end{aligned}$$

Also, we can rewrite $\text{prox}_{\eta v}(\cdot)$ using definition

$$\begin{aligned}
\text{prox}_{\eta v}(x^k - \eta\nabla u(x_g^k)) &= \arg\min_x \left\{\eta v(x) + \frac{1}{2}\|x - (x^k - \eta\nabla u(x_g^k))\|^2\right\} \\
&= \arg\min_x \left\{v(x) + \langle\nabla u(x_g^k), x\rangle + \frac{1}{2\eta}\|x - x^k\|^2\right\}
\end{aligned}$$

and get

$$\begin{aligned}
x_g^k &= \alpha x^k + (1-\alpha)x_f^k \\
\widehat{x}^{k+1} &\approx \arg\min_x \left\{v(x) + \langle\nabla u(x_g^k), x\rangle + \frac{1}{2\eta}\|x - x^k\|^2\right\} \\
x^{k+1} &= x^k - \eta(\nabla u(x_g^k) + \nabla v(\widehat{x}^{k+1})) \\
x_f^{k+1} &= x_g^k + \beta(\widehat{x}^{k+1} - x^k)
\end{aligned} \tag{15}$$

Algorithm 1

Input: $x^0 = x_f^0 \in \mathbb{R}^{d_x}$, $y^0 = y_f^0 \in \mathbb{R}^{d_y}$
Parameters: $\alpha \in (0,1)$, η_x, η_y
for $k = 0, 1, 2, \ldots, K-1$ **do**
  $x_g^k = \alpha x^k + (1-\alpha)x_f^k, \quad y_g^k = \alpha y^k + (1-\alpha)y_f^k$
  $(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \approx \arg\min_{x\in\mathbb{R}^{d_x}} \arg\max_{y\in\mathbb{R}^{d_y}} A_\eta^k(x,y)$

$$A_\eta^k(x,y) := \langle \nabla p(x_g^k), x\rangle + \frac{1}{2\eta_x}\|x - x^k\|^2 + R(x,y) - \langle \nabla q(y_g^k), y\rangle - \frac{1}{2\eta_y}\|y - y^k\|^2 \tag{16}$$

  $x^{k+1} = x^k - \eta_x\left(\nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1})\right)$
  $y^{k+1} = y^k - \eta_y\left(\nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1})\right)$
  $x_f^{k+1} = x_g^k + \alpha(\widehat{x}^{k+1} - x^k)$
  $y_f^{k+1} = y_g^k + \alpha(\widehat{y}^{k+1} - y^k)$
end for
Output: x^K, y^K

The oracle complexity of this method is $\mathcal{O}\left(\sqrt{\frac{L_u}{\mu}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla u(x)$ and $\mathcal{O}\left(\sqrt{\frac{L_v}{\mu}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla v(x)$ to find an ε-solution. Note that this method allow us to separate oracle complexities for composite minimization problem. Now we are ready to get back to composite saddle point problems. Note that for problem (11) is enough to find point $\widehat{x}$ such that $\nabla u(\widehat{x}) + \nabla v(\widehat{x}) = 0$ while for problem (1) we find point $(\widehat{x}, \widehat{y})$ such that $\begin{pmatrix}\nabla p(\widehat{x}) + \nabla_x R(\widehat{x}, \widehat{y}) \\ \nabla q(\widehat{y}) - \nabla_y R(\widehat{x}, \widehat{y})\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$. This fact is allow us to adapt approach (15) to problem (1) with replace $\nabla u(x)$ on operator $A = \begin{pmatrix}\nabla p(x)\\ \nabla q(y)\end{pmatrix}$ and $\nabla v(x)$ on operator $B = \begin{pmatrix}\nabla_x R(x,y) \\ -\nabla_y R(x,y)\end{pmatrix}$ and develop Algorithm 1.

3.2 Complexity

In the following theorem we present the linear convergence of Algorithm 1.

Theorem 1. *Consider Algorithm 1 for solving Problem 1 under Assumptions 1-3, with the following tuning for case* $\frac{L_p}{\mu_x} > \frac{L_q}{\mu_y}$*:*

$$\alpha = \min\left\{1, \sqrt{\frac{\mu_x}{L_p}}\right\} \qquad \eta_x = \min\left\{\frac{1}{3\mu_x}, \frac{1}{3L_p\alpha}\right\} \qquad \eta_y = \frac{\mu_x}{\mu_y}\eta_x \tag{17}$$

or for case $\frac{L_q}{\mu_y} > \frac{L_p}{\mu_x}$*:*

$$\alpha = \min\left\{1, \sqrt{\frac{\mu_y}{L_q}}\right\} \qquad \eta_y = \min\left\{\frac{1}{3\mu_y}, \frac{1}{3L_q\alpha}\right\} \qquad \eta_x = \frac{\mu_y}{\mu_x}\eta_y \tag{18}$$

and let $(\widehat{x}^{k+1}, \widehat{y}^{k+1})$ *in algorithm 1 satisfy*

$$\eta_x\|\nabla_x A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1})\|^2 + \eta_y\|\nabla_y A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1})\|^2 \le \frac{1}{6\eta_x}\|\widehat{x}^{k+1} - x^k\|^2 + \frac{1}{6\eta_y}\|\widehat{y}^{k+1} - y^k\|^2 \tag{19}$$

Then, for any

$$K \geq 3 \max \left\{ 1, \sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}} \right\} \log \frac{\frac{1}{\eta_x}\|x^0 - x^*\|^2 + \frac{1}{\eta_y}\|y^0 - y^*\|^2 + \frac{2}{\alpha}\mathrm{D}_p(x_f^0, x^*) + \frac{2}{\alpha}\mathrm{D}_q(y_f^0, y^*)}{\varepsilon} \tag{20}$$

we have the following estimate for the distance to the solution (x^*, y^*)*:*

$$\frac{1}{\eta_x}\|x^K - x^*\|^2 + \frac{1}{\eta_y}\|y^K - y^*\|^2 \leq \varepsilon \tag{21}$$

Proof of Theorem 1 you can find in Section 4.1.

Auxiliary subproblem complexity. At each iteration of Algorithm 1 we need to find $\widehat{x}^{k+1}, \widehat{y}^{k+1}$ (solution to the problem (16)) that satisfies condition (19). $A_\eta^k(x, y)$ is $\left(L_R + \frac{1}{\eta_x}\right)$-smooth in x for fixed y and $\left(L_R + \frac{1}{\eta_y}\right)$-smooth in y for fixed x. Due to this property we get the following inequality

$$\begin{aligned} &\eta_x \|\nabla_x A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1})\|^2 + \eta_y \|\nabla_y A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1})\|^2 \\ &\quad \leq \max\left\{ \eta_x \left(L_R + \frac{1}{\eta_x} \right)^2, \eta_y \left(L_R + \frac{1}{\eta_y} \right)^2 \right\} \left(\|\widehat{x}^{k+1} - \widehat{x}_*^{k+1}\|^2 + \|\widehat{y}^{k+1} - \widehat{y}_*^{k+1}\|^2 \right) \end{aligned}$$

where $(\widehat{x}_*^{k+1}, \widehat{y}_*^{k+1})$ is the solution to the problem (16). It means that $\frac{\frac{1}{6\eta_x}\|\widehat{x}^{k+1} - x^k\|^2 + \frac{1}{6\eta_y}\|\widehat{y}^{k+1} - y^k\|^2}{\max\left\{\eta_x\left(L_R + \frac{1}{\eta_x}\right)^2, \eta_y\left(L_R + \frac{1}{\eta_y}\right)^2\right\}}$-solution to (16) satisfies condition (19). To find this solution we can apply the algorithm FOAM (Algorithm 4 from [KG22]) from starting point (x^k, y^k) and get the following complexity.

Theorem 2. *Algorithm 4 from [KG22] requires the following number of gradient evaluations:*

$$T = \mathcal{O}\left(\left(1 + L_R\sqrt{\eta_x \eta_y}\right) \log \frac{1}{\gamma} \right) \tag{22}$$

to find an γ*-accurate solution of problem* (16) *with*

$$\gamma = \frac{\frac{1}{6\eta_x}\|\widehat{x}^{k+1} - x^k\|^2 + \frac{1}{6\eta_y}\|\widehat{y}^{k+1} - y^k\|^2}{\max\left\{ \eta_x \left(L_R + \frac{1}{\eta_x}\right)^2, \eta_y \left(L_R + \frac{1}{\eta_y}\right)^2 \right\}} \tag{23}$$

Proof of Theorem 2 you can find in Section 4.2.

Remark. Note that the stopping criterion (19) for solving the auxiliary problem (16) is practical due to it does not depend on point $(\widehat{x}_*^{k+1}, \widehat{y}_*^{k+1})$ (solution to (16)).

Overall complexity. To formulate the total iterative complexity of Algorithm 1 we do some mathematical calculations for case $\frac{L_p}{\mu_x} \geq \frac{L_q}{\mu_y}$.

$$K \times T = \mathcal{O}\left(\left(1 + \sqrt{\frac{L_p}{\mu_x}} \right) \log \frac{1}{\varepsilon} \right) \times \mathcal{O}\left(\left(1 + L_R\sqrt{\eta_x \eta_y}\right) \log \frac{1}{\gamma} \right)$$

$$
\begin{aligned}
&= \mathcal{O}\left(\left(1 + \sqrt{\frac{L_p}{\mu_x}} + L_R\sqrt{\frac{L_p}{\mu_y}}\eta_x\right)\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right) \\
&= \mathcal{O}\left(\left(1 + \sqrt{\frac{L_p}{\mu_x}} + \frac{L_R}{\sqrt{\mu_x\mu_y}}\right)\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)
\end{aligned}
$$

The case $\frac{L_q}{\mu_y} > \frac{L_p}{\mu_x}$ is symmetric. Solving the auxiliary subproblem (16) does not require calling oracles $\nabla p(x), \nabla q(y)$. These oracles are called only in algorithm 1 of Algorithm 1. Summing up the oracle complexity of Algorithm 1 we present in the following theorem.

Theorem 3. *Consider Problem* (1) *under Assumptions 1 to 3. Then, to find an ε-solution, Algorithm 1 requires*

$$
\mathcal{O}\left(\max\left\{1, \sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right) \quad \textit{calls of } \; \nabla p(x), \nabla q(y)
$$

and

$$
\mathcal{O}\left(\left(\sqrt{\frac{L_p}{\mu_x}} + \sqrt{\frac{L_q}{\mu_y}} + \frac{L_R}{\sqrt{\mu_x\mu_y}}\right)\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right) \quad \textit{calls of } \; \nabla R(x,y)
$$

SPP with one composite. The important particular case of problem (1) is composite saddle point problem with one composite. It means that in this case $L_q = 0$ (or $L_p = 0$). By Theorem 3 Algorithm 1 requires $\mathcal{O}\left(\sqrt{\frac{L_p}{\mu_x}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla p(x)$ and $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \frac{L_R}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla R(x,y)$ to find an ε-solution to problem (1).

3.3 Discussion Our Results and Related Works

In this subsection we discuss the lower bounds and compare iteration and oracle complexities results for Algorithm 1 stated in Theorem 3 with related works.

Lower bounds. The lower bounds on iteration complexity to the *strongly-convex-strongly-concave* problem (1) is $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \frac{L_R}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$. This result was presented in [ZHZ19]. The special case of the problem (1) is

$$
\min_x \max_y \frac{1}{2}\|x\|^2 + R(x,y) - \frac{1}{2}\|y\|^2
$$

This means that the lower bounds on the oracle calls of $\nabla R(x,y)$ is $\Omega\left(\frac{L_R}{\sqrt{\mu_x\mu_y}}\log\frac{1}{\varepsilon}\right)$. Also, the problem (1) has a special case

$$
\min_x \max_y p(x) + \frac{\mu_x}{2}\|x\|^2 - q(y) - \frac{\mu_y}{2}\|y\|^2
$$

that separate into two problems $\min_x p(x) + \frac{\mu_x}{2}\|x\|^2$ and $\max_y -q(y) - \frac{\mu_y}{2}\|y\|^2$. The lower bounds to these problems on oracle calls of $\nabla p(x), \nabla q(y)$ is $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$ which was proposed in [Nes18]. To sum up, the oracle complexities to the problem (1) is $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla p(x), \nabla q(y)$ and $\Omega\left(\frac{L_R}{\sqrt{\mu_x\mu_y}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla R(x,y)$.

Strongly-convex-strongly-concave case. For the strongly-convex-strongly-concave case Algorithm 1 has the following oracle complexity

$$\mathcal{O}\left(\max\left\{1, \sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right) \quad \text{calls of} \quad \nabla p(x), \nabla q(y)$$

and

$$\mathcal{O}\left(\frac{L_R}{\sqrt{\mu_x\mu_y}}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right) \quad \text{calls of} \quad \nabla R(x,y)$$

to find an ε-solution to (1). Also, the iteration complexity of Algorithm 1 is $\mathcal{O}\left(\max\left\{1, \sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \frac{L_R}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)$. This achieves the lower bounds up to logarithmic factor and improves the results for iteration complexity

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \sqrt{\frac{L_R\max\{L_p,L_q,L_R\}}{\mu_x\mu_y}}\right\}\log^3\frac{(L_p+L_R)(L_q+L_R)}{\mu_x\mu_y}\log\frac{1}{\varepsilon}\right)$$

from [WL20],

$$\mathcal{O}\left(\frac{L_R+\sqrt{L_pL_q}}{\sqrt{\mu_x\mu_y}}\log^3\frac{1}{\varepsilon}\right)$$

according to [LJJ20],

$$\mathcal{O}\left(\frac{L_R+\sqrt{L_pL_q}}{\sqrt{\mu_x\mu_y}}\log\frac{1}{\varepsilon}\right)$$

from [KG22] and

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \frac{L_R}{\mu_x}, \frac{L_R}{\mu_y}\right\}\log\frac{1}{\varepsilon}\right)$$

according to [JST22]. Note, that in works [JST22] considered problem (1) under Assumptions 1, 2 and the following assumption on function $R(x,y)$.

Assumption 4. *$R(x,y)$ is twice differentiable function and $\|\nabla_{xx}R(x,y)\| \le L_R^{xx}$, $\|\nabla_{yy}R(x,y)\| \le L_R^{yy}$ and $\|\nabla_{xy}R(x,y)\| \le L_R^{xy}$, where $\|\cdot\|$ is spectral norm.*

Using these notation the authors of [JST22] get the following iteration complexity to problem (1):

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \frac{L_R^{xx}}{\mu_x}, \frac{L_R^{yy}}{\mu_y}, \frac{L_R^{xy}}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$$

This iteration complexity achieve the lower bounds if $L_R^{xx} = L_R^{yy} = 0$. Meanwhile, Algorithm 1 effectively (achieves the lower bounds) separates the oracle calls for composite functions $\nabla p(x), \nabla q(y)$ and for saddle part $\nabla R(x,y)$ up to logarithmic factor.

In work [ADS$^+$19], the authors separate the oracle calls for $\nabla p(x)$, $\nabla q(y)$ and $\nabla R(x,y)$ but these bounds not achieves the lower bounds even for iteration complexity. Due to these facts, to the best of our knowledge, Algorithm 1 is the first algorithm that achieves the lower bounds on iteration and separate effectively the oracle calls to (1).

4 Missing proofs

4.1 Proof of Theorem 1

Lemma 1. *Under Assumptions 1-3, the following inequality holds for Algorithm 1.*

$$
\begin{aligned}
&-2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle \\
\le &-\frac{2}{\alpha}\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} x_f^{k+1} - x_g^k \\ y_f^{k+1} - y_g^k \end{pmatrix} \right\rangle + \frac{2(1-\alpha)}{\alpha}(\mathrm{D}_p(x_f^k, x^*) - \mathrm{D}_p(x_g^k, x^*)) \\
&+ \frac{2(1-\alpha)}{\alpha}(\mathrm{D}_q(y_f^k, y^*) - \mathrm{D}_q(y_g^k, y^*)) - 2\mathrm{D}_p(x_g^k; x^*) - 2\ \mathrm{D}_q(y_g^k; y^*) \\
&- \mu_x \|\widehat{x}^{k+1} - x^*\|^2 - \mu_y \|\widehat{y}^{k+1} - y^*\|^2
\end{aligned}
$$

Proof. Using the first-order necessary condition $\begin{pmatrix} \nabla p(x^*) \\ \nabla q(y^*) \end{pmatrix} + \begin{pmatrix} \nabla_x R(x^*, y^*) \\ -\nabla_y R(x^*, y^*) \end{pmatrix} = 0$, μ_x-strong convexity in x and μ_y-strong concavity in y of $R(x, y)$, we get

$$
\begin{aligned}
&-2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle \\
= &-2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) - \nabla p(x^*) - \nabla_x R(x^*, y^*) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) - \nabla q(y^*) + \nabla_y R(x^*, y^*) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle \\
\le &-2\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle - \mu_x \|\widehat{x}^{k+1} - x^*\|^2 - \mu_y \|\widehat{y}^{k+1} - y^*\|^2 \\
= &-2\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^k \\ \widehat{y}^{k+1} - y^k \end{pmatrix} \right\rangle - 2\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} x^k - x_g^k \\ y^k - y_g^k \end{pmatrix} \right\rangle \\
&-2\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} x_g^k - x^* \\ y_g^k - y^* \end{pmatrix} \right\rangle - \mu_x \|\widehat{x}^{k+1} - x^*\|^2 - \mu_y \|\widehat{y}^{k+1} - y^*\|^2
\end{aligned}
$$

Using convexity of $p(x)$ and $q(y)$, we get

$$
\begin{aligned}
&-2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle \\
\le &-2\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^k \\ \widehat{y}^{k+1} - y^k \end{pmatrix} \right\rangle \\
&-2\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} x^k - x_g^k \\ y^k - y_g^k \end{pmatrix} \right\rangle \\
&- 2\mathrm{D}_p(x_g^k; x^*) - 2\ \mathrm{D}_q(y_g^k; y^*) \\
&- \mu_x \|\widehat{x}^{k+1} - x^*\|^2 - \mu_y \|\widehat{y}^{k+1} - y^*\|^2
\end{aligned}
$$

Now, we use algorithm 1 and algorithm 1 of Algorithm 1 and get

$$
-2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle
$$

$$
\begin{aligned}
&\leq -\frac{2}{\alpha}\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} x_f^{k+1} - x_g^k \\ y_f^{k+1} - y_g^k \end{pmatrix} \right\rangle \\
&\quad + \frac{2(1-\alpha)}{\alpha}\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} x_f^{k} - x_g^k \\ y_f^{k} - y_g^k \end{pmatrix} \right\rangle \\
&\quad - 2\mathrm{D}_p(x_g^k; x^*) - 2\ \mathrm{D}_q(y_g^k; y^*) \\
&\quad - \mu_x\|\widehat{x}^{k+1} - x^*\|^2 - \mu_y\|\widehat{y}^{k+1} - y^*\|^2 \\
&= -\frac{2}{\alpha}\left\langle \begin{pmatrix} \nabla p(x_g^k) - \nabla p(x^*) \\ \nabla q(y_g^k) - \nabla q(y^*) \end{pmatrix}; \begin{pmatrix} x_f^{k+1} - x_g^k \\ y_f^{k+1} - y_g^k \end{pmatrix} \right\rangle \\
&\quad + \frac{2(1-\alpha)}{\alpha}(\mathrm{D}_p(x_f^k, x^*) - \mathrm{D}_p(x_g^k, x^*)) \\
&\quad + \frac{2(1-\alpha)}{\alpha}(\mathrm{D}_q(y_f^k, y^*) - \mathrm{D}_q(y_g^k, y^*)) \\
&\quad - 2\mathrm{D}_p(x_g^k; x^*) - 2\mathrm{D}_q(y_g^k; y^*) \\
&\quad - \mu_x\|\widehat{x}^{k+1} - x^*\|^2 - \mu_y\|\widehat{y}^{k+1} - y^*\|^2
\end{aligned}
$$

This completes the proof of Lemma. □

Assumption 5. $\frac{L_p}{\mu_x} \geq \frac{L_q}{\mu_y}$

Lemma 2. *Consider Algorithm 1 for Problem 1 under Assumptions 1-5, with the following tuning:*

$$
\alpha = \min\left\{1, \sqrt{\frac{\mu_x}{L_p}}\right\}, \quad \eta_x = \min\left\{\frac{1}{3\mu_x}, \frac{1}{3L_p\alpha}\right\}, \quad \eta_y = \frac{\mu_x}{\mu_y}\eta_x \tag{24}
$$

and let $\widehat{x}^{k+1}$ in algorithm 1 satisfy

$$
\eta_x\|\nabla_x A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1})\|^2 + \eta_y\|\nabla_y A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1})\|^2 \leq \frac{1}{6\eta_x}\|\widehat{x}^{k+1} - x^k\|^2 + \frac{1}{6\eta_y}\|\widehat{y}^{k+1} - y^k\|^2 \tag{25}
$$

Then, the following inequality holds:

$$
\Psi^{k+1} \leq \left(1 - \frac{\alpha}{3}\right)\Psi^k \tag{26}
$$

where

$$
\Psi^k := \frac{1}{\eta_x}\|x^k - x^*\|^2 + \frac{1}{\eta_y}\|y^k - y^*\|^2 + \frac{2}{\alpha}\mathrm{D}_p(x_f^k, x^*) + \frac{2}{\alpha}\mathrm{D}_q(y_f^k, y^*) \tag{27}
$$

Proof. Using algorithm 1 of Algorithm 1, we get

$$
\begin{aligned}
\left\|\begin{matrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{matrix}\right\|_P^2 &= \left\|\begin{matrix} x^{k} - x^* \\ y^{k} - y^* \end{matrix}\right\|_P^2 + 2\left\langle \begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix}; \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \right\rangle_P + \left\|\begin{matrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{matrix}\right\|_P^2 \\
&= \left\|\begin{matrix} x^{k} - x^* \\ y^{k} - y^* \end{matrix}\right\|_P^2 - 2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \right\rangle + \left\|\begin{matrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{matrix}\right\|_P^2 \\
&= \left\|\begin{matrix} x^{k} - x^* \\ y^{k} - y^* \end{matrix}\right\|_P^2 + 2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \frac{\widehat{x}^{k+1} - x^k}{\eta_x} \\ \frac{\widehat{y}^{k+1} - y^k}{\eta_y} \end{pmatrix} \right\rangle_{P^{-1}}
\end{aligned}
$$

$$- 2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle + \left\| \begin{matrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{matrix} \right\|_P^2$$

Since $2\langle a, b\rangle = \|a+b\|^2 - \|a\|^2 - \|b\|^2$, we get

$$\begin{aligned} \left\| \begin{matrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{matrix} \right\|_P^2 &= \left\| \begin{matrix} x^k - x^* \\ y^k - y^* \end{matrix} \right\|_P^2 + \left\| \begin{matrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) + \frac{\widehat{x}^{k+1} - x^k}{\eta_x} \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) + \frac{\widehat{y}^{k+1} - y^k}{\eta_y} \end{matrix} \right\|_{P^{-1}}^2 \\ &\quad - \left\| \begin{matrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{matrix} \right\|_{P^{-1}}^2 - \left\| \begin{matrix} \widehat{x}^{k+1} - x^k \\ \widehat{y}^{k+1} - y^k \end{matrix} \right\|_P^2 \\ &\quad - 2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle + \left\| \begin{matrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{matrix} \right\|_P^2 \end{aligned}$$

Using algorithm 1 of Algorithm 1, we get

$$\begin{aligned} \left\| \begin{matrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{matrix} \right\|_P^2 &= \left\| \begin{matrix} x^k - x^* \\ y^k - y^* \end{matrix} \right\|_P^2 + \left\| \begin{matrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) + \frac{\widehat{x}^{k+1} - x^k}{\eta_x} \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) + \frac{\widehat{y}^{k+1} - y^k}{\eta_y} \end{matrix} \right\|_{P^{-1}}^2 - \left\| \begin{matrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{matrix} \right\|_P^2 - \left\| \begin{matrix} \widehat{x}^{k+1} - x^k \\ \widehat{y}^{k+1} - y^k \end{matrix} \right\|_P^2 \\ &\quad - 2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle + \left\| \begin{matrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{matrix} \right\|_P^2 \\ &= \left\| \begin{matrix} x^k - x^* \\ y^k - y^* \end{matrix} \right\|_P^2 + 2\left\| \begin{matrix} \nabla_x A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ -\nabla_y A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{matrix} \right\|_{P^{-1}}^2 - \left\| \begin{matrix} \widehat{x}^{k+1} - x^k \\ \widehat{y}^{k+1} - y^k \end{matrix} \right\|_P^2 \\ &\quad - 2\left\langle \begin{pmatrix} \nabla p(x_g^k) + \nabla_x R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ \nabla q(y_g^k) - \nabla_y R(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{pmatrix}; \begin{pmatrix} \widehat{x}^{k+1} - x^* \\ \widehat{y}^{k+1} - y^* \end{pmatrix} \right\rangle \end{aligned}$$

Using Lemma 2 and algorithm 1 of Algorithm 1 , we get

$$\begin{aligned} \left\| \begin{matrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{matrix} \right\|_P^2 &\le \left\| \begin{matrix} x^k - x^* \\ y^k - y^* \end{matrix} \right\|_P^2 + 2\left\| \begin{matrix} \nabla_x A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ -\nabla_y A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{matrix} \right\|_{P^{-1}}^2 - \frac{2}{3}\left\| \begin{matrix} \widehat{x}^{k+1} - x^k \\ \widehat{y}^{k+1} - y^k \end{matrix} \right\|_P^2 \\ &\quad - \frac{2}{\alpha}\left(\left\langle \nabla p(x_g^k) - \nabla p(x^*); x_f^{k+1} - x_g^k \right\rangle + \frac{1}{6\alpha\eta_x}\|x_f^{k+1} - x_g^k\|^2 \right) \\ &\quad - \frac{2}{\alpha}\left(\left\langle \nabla q(y_g^k) - \nabla q(y^*); y_f^{k+1} - y_g^k \right\rangle + \frac{1}{6\alpha\eta_y}\|y_f^{k+1} - y_g^k\|^2 \right) \\ &\quad + \frac{2(1-\alpha)}{\alpha}(\mathrm{D}_p(x_f^k, x^*) - \mathrm{D}_p(x_g^k, x^*)) + \frac{2(1-\alpha)}{\alpha}(\mathrm{D}_q(y_f^k, y^*) - \mathrm{D}_q(y_g^k, y^*)) \\ &\quad - 2\mathrm{D}_p(x_g^k; x^*) - 2\,\mathrm{D}_q(y_g^k; y^*) - \mu_x\|\widehat{x}^{k+1} - x^*\|^2 - \mu_y\|\widehat{y}^{k+1} - y^*\|^2 \end{aligned}$$

Since $\eta_x \le \frac{1}{3L_p\alpha}$, $\eta_y \le \frac{\mu_x}{\mu_y}\frac{1}{3L_p\alpha} \le \frac{1}{3L_q\alpha}$ (by (24) and Assumption 5)

$$\begin{aligned} \left\| \begin{matrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{matrix} \right\|_P^2 &\le \left\| \begin{matrix} x^k - x^* \\ y^k - y^* \end{matrix} \right\|_P^2 + 2\left\| \begin{matrix} \nabla_x A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \\ -\nabla_y A_\eta^k(\widehat{x}^{k+1}, \widehat{y}^{k+1}) \end{matrix} \right\|_{P^{-1}}^2 - \frac{2}{3}\left\| \begin{matrix} \widehat{x}^{k+1} - x^k \\ \widehat{y}^{k+1} - y^k \end{matrix} \right\|_P^2 \\ &\quad - \frac{2}{\alpha}\left(\left\langle \nabla p(x_g^k) - \nabla p(x^*); x_f^{k+1} - x_g^k \right\rangle + \frac{L_p}{2}\|x_f^{k+1} - x_g^k\|^2 \right) \\ &\quad - \frac{2}{\alpha}\left(\left\langle \nabla q(y_g^k) - \nabla q(y^*); y_f^{k+1} - y_g^k \right\rangle + \frac{L_q}{2}\|y_f^{k+1} - y_g^k\|^2 \right) \end{aligned}$$

$$
\begin{aligned}
&+\frac{2(1-\alpha)}{\alpha}(\mathrm{D}_p(x_f^k,x^*)-\mathrm{D}_p(x_g^k,x^*))+\frac{2(1-\alpha)}{\alpha}(\mathrm{D}_q(y_f^k,y^*)-\mathrm{D}_q(y_g^k,y^*))\\
&-2\mathrm{D}_p(x_g^k;x^*)-2\ \mathrm{D}_q(y_g^k;y^*)-\mu_x\|\widehat{x}^{k+1}-x^*\|^2-\mu_y\|\widehat{y}^{k+1}-y^*\|^2
\end{aligned}
$$

L_p-smoothness of $p(x)$ and L_q-smoothness of $q(y)$ gives

$$
\begin{aligned}
\left\|\begin{matrix}x^{k+1}-x^*\\ y^{k+1}-y^*\end{matrix}\right\|_P^2 \le{}& \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_P^2 + 2\left\|\begin{matrix}\nabla_x A_\eta^k(\widehat{x}^{k+1},\widehat{y}^{k+1})\\ -\nabla_y A_\eta^k(\widehat{x}^{k+1},\widehat{y}^{k+1})\end{matrix}\right\|_{P^{-1}}^2 - \frac{2}{3}\left\|\begin{matrix}\widehat{x}^{k+1}-x^k\\ \widehat{y}^{k+1}-y^k\end{matrix}\right\|_P^2\\
&-\frac{2}{\alpha}(\mathrm{D}_p(x_f^{k+1},x^*)-\mathrm{D}_p(x_g^k,x^*))-\frac{2}{\alpha}(\mathrm{D}_q(y_f^{k+1},y^*)-\mathrm{D}_q(y_g^k,y^*))\\
&+\frac{2(1-\alpha)}{\alpha}(\mathrm{D}_p(x_f^k,x^*)-\mathrm{D}_p(x_g^k,x^*))+\frac{2(1-\alpha)}{\alpha}(\mathrm{D}_q(y_f^k,y^*)-\mathrm{D}_q(y_g^k,y^*))\\
&-2\mathrm{D}_p(x_g^k;x^*)-2\ \mathrm{D}_q(y_g^k;y^*)-\mu_x\|\widehat{x}^{k+1}-x^*\|^2-\mu_y\|\widehat{y}^{k+1}-y^*\|^2\\
={}&\left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_P^2 + 2\Big(\eta_x\|\nabla_x A_\eta^k(\widehat{x}^{k+1},\widehat{y}^{k+1})\|^2+\eta_y\|\nabla_y A_\eta^k(\widehat{x}^{k+1},\widehat{y}^{k+1})\|^2\\
&-\frac{1}{6\eta_x}\|\widehat{x}^{k+1}-x^k\|^2-\frac{1}{6\eta_y}\|\widehat{y}^{k+1}-y^k\|^2\Big)-\frac{2}{\alpha}\mathrm{D}_p(x_f^{k+1},x^*)-\frac{2}{\alpha}\mathrm{D}_q(y_f^{k+1},y^*)\\
&+\frac{2(1-\alpha)}{\alpha}\mathrm{D}_p(x_f^k,x^*)+\frac{2(1-\alpha)}{\alpha}\mathrm{D}_q(y_f^k,y^*)\\
&-\mu_x\left(\|\widehat{x}^{k+1}-x^*\|^2+\frac{1}{3\mu_x\eta_x}\|\widehat{x}^{k+1}-x^k\|^2\right)-\mu_y\left(\|\widehat{y}^{k+1}-y^*\|^2+\frac{1}{3\eta_y\mu_y}\|\widehat{y}^{k+1}-y^k\|^2\right)
\end{aligned}
$$

Since $\eta_x \le \frac{1}{3\mu_x}$, $\eta_y = \frac{\mu_x}{\mu_y}\eta_x \le \frac{1}{3\mu_y}$ (by (24)). Using inequality $-\|a-b\|^2 \ge -2\|a\|^2-2\|b\|^2$, we get

$$
\begin{aligned}
\left\|\begin{matrix}x^{k+1}-x^*\\ y^{k+1}-y^*\end{matrix}\right\|_P^2 \le{}& \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_P^2 + 2\Big(\eta_x\|\nabla_x A_\eta^k(\widehat{x}^{k+1},\widehat{y}^{k+1})\|^2+\eta_y\|\nabla_y A_\eta^k(\widehat{x}^{k+1},\widehat{y}^{k+1})\|^2-\frac{1}{6\eta_x}\|\widehat{x}^{k+1}-x^k\|^2\\
&-\frac{1}{6\eta_y}\|\widehat{y}^{k+1}-y^k\|^2\Big)-\frac{2}{\alpha}\mathrm{D}_p(x_f^{k+1},x^*)-\frac{2}{\alpha}\mathrm{D}_q(y_f^{k+1},y^*)+\frac{2(1-\alpha)}{\alpha}\mathrm{D}_p(x_f^k,x^*)\\
&+\frac{2(1-\alpha)}{\alpha}\mathrm{D}_q(y_f^k,y^*)-\mu_x\|x^k-x^*\|^2-\mu_y\|y^k-y^*\|^2
\end{aligned}
$$

Since (25), we get

$$
\begin{aligned}
\frac{1}{\eta_x}\|x^{k+1}-x^*\|^2+&\frac{1}{\eta_y}\|y^{k+1}-y^*\|^2+\frac{2}{\alpha}\mathrm{D}_p(x_f^{k+1},x^*)+\frac{2}{\alpha}\mathrm{D}_q(y_f^{k+1},y^*)\le\\
&\le\frac{1}{\eta_x}\left(1-\mu_x\eta_x\right)\|x^k-x^*\|^2+\frac{1}{\eta_y}\left(1-\mu_y\eta_y\right)\|y^k-y^*\|^2\\
&\quad+\frac{2(1-\alpha)}{\alpha}\mathrm{D}_p(x_f^k,x^*)+\frac{2(1-\alpha)}{\alpha}\mathrm{D}_q(y_f^k,y^*)\\
&\le(1-\alpha)\left[\frac{1}{\eta_x}\|x^k-x^*\|^2+\frac{1}{\eta_y}\|y^k-y^*\|^2+\frac{2}{\alpha}\mathrm{D}_p(x_f^k,x^*)+\frac{2}{\alpha}\mathrm{D}_q(y_f^k,y^*)\right]
\end{aligned}
$$

In the last inequality we use that $\alpha > \frac{\alpha}{3}$, $\eta_x\mu_x \ge \frac{\alpha}{3}$ and $\eta_y\mu_y \ge \frac{\alpha}{3}$. If $L_p \le \mu_x$, then $\alpha = 1$, $\eta_x\mu_x = \frac{1}{3} = \frac{\alpha}{3}$, $\eta_y\mu_y = \frac{1}{3} = \frac{\alpha}{3}$. If $L_p > \mu_x$, then $\alpha = \sqrt{\frac{\mu_x}{L_p}}$, $\eta_x\mu_x = \sqrt{\frac{\mu_x}{3L_p}} \ge \frac{\alpha}{3}$, $\eta_y\mu_y = \sqrt{\frac{\mu_x}{3L_p}} \ge \frac{\alpha}{3}$.

By (27) definition of Ψ^k, we get

$$\Psi^{k+1} \leq \left(1 - \frac{\alpha}{3}\right) \Psi^k$$

$\square$

Proof of Theorem 1 Using the property of the Bregman devergence $\mathrm{D}_f(x, y) \geq 0$ and running the recursion (26) we get

$$\frac{1}{\eta_x}\|x^K - x^*\|^2 + \frac{1}{\eta_y}\|y^K - y^*\|^2 \leq \Psi^K \leq C\left(1 - \frac{\alpha}{3}\right)^K$$

where C is defined as

$$C = \frac{1}{\eta_x}\|x^0 - x^*\|^2 + \frac{1}{\eta_y}\|y^0 - y^*\|^2 + \frac{2}{\alpha}\mathrm{D}_p(x_f^0, x^*) + \frac{2}{\alpha}\mathrm{D}_q(y_f^0, y^*)$$

After $K = \frac{3}{\alpha}\log\frac{1}{\varepsilon}$ iterations of Algorithm 1 we get a pair (x^K, y^K) satisfies the following inequality

$$\frac{1}{\eta_x}\|x^K - x^*\|^2 + \frac{1}{\eta_y}\|y^K - y^*\|^2 \leq \varepsilon$$

4.2 Proof of Theorem 2

Lemma 3. *Consider function $R(x, y)$ under Assumption 3. If we make the following replacing variables $x = \alpha u$, $y = \beta v$, then function $\widetilde{R}(u, v) := R(x, y)$ is $\widetilde{L}$-smooth, μ_u-strongly convex in u with fixed v and μ_v-strongly concave in v for fixed u, with $\widetilde{L} = \max\{\alpha^2, \beta^2\}L$, $\mu_u = \alpha^2\mu_x$, $\mu_v = \beta^2\mu_y$*

Proof. Firstly, let us consider that

$$\nabla_x R(x, y) = \begin{pmatrix} \frac{\partial R(x,y)}{\partial x_1} \\ \dots \\ \frac{\partial R(x,y)}{\partial x_{d_x}} \end{pmatrix} = \begin{pmatrix} \frac{\partial R(\alpha u,y)}{\partial u_1} \cdot \frac{\partial u_1}{\partial x_1} \\ \dots \\ \frac{\partial R(\alpha u,y)}{\partial u_{d_x}} \cdot \frac{\partial u_{d_x}}{\partial x_{d_x}} \end{pmatrix} = \begin{pmatrix} \frac{\partial R(\alpha u,y)}{\partial u_1} \cdot \frac{1}{\alpha} \\ \dots \\ \frac{\partial R(\alpha u,y)}{\partial u_{d_x}} \cdot \frac{1}{\alpha} \end{pmatrix} = \frac{1}{\alpha}\nabla_u \widetilde{R}(u, v)$$

Using the analogical calculations we get $\nabla_y R(x, y) = \frac{1}{\beta}\nabla_v \widetilde{R}(u, v)$. Now we define the smoothness constant of function $\widetilde{R}(u, v)$ using L-smoothness of function $R(x, y)$.

$$\begin{aligned}
\|\nabla\widetilde{R}(u_1, v_1) - \nabla\widetilde{R}(u_2, v_2)\|^2 &= \|\nabla_u\widetilde{R}(u_1, v_1) - \nabla_u\widetilde{R}(u_2, v_2)\|^2 + \|\nabla_v\widetilde{R}(u_1, v_1) - \nabla_v\widetilde{R}(u_2, v_2)\|^2 \\
&= \alpha^2\|\nabla_x R(x_1, y_1) - \nabla_x R(x_2, y_2)\|^2 + \beta^2\|\nabla_y R(x_1, y_1) - \nabla_y R(x_2, y_2)\|^2 \\
&\leq \max\{\alpha^2, \beta^2\}\|\nabla R(x_1, y_1) - \nabla R(x_2, y_2)\|^2 \\
&\leq \max\{\alpha^2, \beta^2\}L^2\left(\|x_1 - x_2\|^2 + \|y_1 - y_2\|^2\right) \\
&= \max\{\alpha^2, \beta^2\}L^2\left(\alpha^2\|u_1 - u_2\|^2 + \beta^2\|v_1 - v_2\|^2\right) \\
&\leq \widetilde{L}^2\left(\|u_1 - u_2\|^2 + \|v_1 - v_2\|^2\right)
\end{aligned}$$

with $\widetilde{L} = \max\{\alpha^2, \beta^2\}L$.

Now we define μ_u-strongly convex constant of function $\widetilde{R}(u, v)$ in u for fixed v.

$$\widetilde{R}(u_2, v) = R(x_2, y) \geq R(x_1, y) + \langle\nabla_x R(x_1, y), x_2 - x_1\rangle + \frac{\mu_x}{2}\|x_2 - x_1\|^2$$

$$
\begin{aligned}
&= \widetilde{R}(u_1, v) + \left\langle \frac{1}{\alpha}\nabla_u \widetilde{R}(u_1, v), \alpha(u_2 - u_1)\right\rangle + \frac{\mu_x \alpha^2}{2}\|u_2 - u_1\|^2 \\
&= \widetilde{R}(u_1, v) + \left\langle \nabla_u \widetilde{R}(u_1, v), u_2 - u_1\right\rangle + \frac{\mu_u}{2}\|u_2 - u_1\|^2
\end{aligned}
$$

with $\mu_u = \alpha^2\mu_x$. In this equation we use μ_x-strong convexity of $R(x, y)$ in x for fixed y and differentiation rule of complex function. Similarly we get μ_v-strong concavity of $\widetilde{R}(u, v)$ in v for fixed u, with $\mu_v = \beta^2\mu_y$. □

Proof of Theorem 2 Firstly, we make the following replacing variables $x = \alpha u$, $y = \beta v$ in the problem (16). After that we get the following problem in new variables:

$$
\min_u \max_v \alpha\langle\nabla p(x_g^k), u\rangle + \frac{\alpha^2}{2\eta_x}\left\|u - \frac{x^k}{\alpha}\right\|^2 + \widetilde{R}(u, v) - \beta\langle\nabla q(y_g^k), v\rangle - \frac{\beta^2}{2\eta_y}\left\|v - \frac{y^k}{\beta}\right\|^2 \tag{28}
$$

By Corollary 1 from [KG22] Algorithm FOAM (Algorithm 4 from [KG22]) requires the following number of gradient evaluations:

$$
T = \mathcal{O}\left(\left(\frac{\widetilde{L}_R + \frac{\alpha^2}{\eta_x} + \frac{\beta^2}{\eta_y}}{\sqrt{(\mu_u + \frac{\alpha^2}{\eta_x})(\mu_v + \frac{\beta^2}{\eta_y})}}\right)\log\frac{1}{\gamma}\right) \tag{29}
$$

to find an γ-accurate solution of problem (28). By Lemma 3 we get $\widetilde{L}_R = \max\{\alpha, \beta\}L_R$, $\mu_u = \alpha^2\mu_x$ and $\mu_v = \beta^2\mu_y$. Using these values we get the following number of gradient evaluations:

$$
T = \mathcal{O}\left(\left(\frac{\max\{\alpha^2, \beta^2\}L_R + \frac{\alpha^2}{\eta_x} + \frac{\beta^2}{\eta_y}}{\sqrt{(\alpha^2\mu_x + \frac{\alpha^2}{\eta_x})(\beta^2\mu_y + \frac{\beta^2}{\eta_y})}}\right)\log\frac{1}{\gamma}\right) \tag{30}
$$

Now we are ready to define constants α, β. For case $\eta_x > \eta_y$ we define $\alpha^2 = \sqrt{\frac{\eta_x}{\eta_y}}$, $\beta = 1$. For another case ($\eta_x \le \eta_y$) we define $\alpha = 1$, $\beta^2 = \sqrt{\frac{\eta_y}{\eta_x}}$. We provide proof only for case $\eta_x > \eta_y$ due to case $\eta_x \le \eta_y$ is symmetric.

$$
\begin{aligned}
T = \mathcal{O}\left(\left(\frac{\sqrt{\frac{\eta_x}{\eta_y}}L_R + \frac{2}{\eta_y}}{\sqrt{\left(\sqrt{\frac{\eta_x}{\eta_y}}\mu_x + \frac{1}{\eta_y}\right)\left(\mu_y + \frac{1}{\eta_y}\right)}}\right)\log\frac{1}{\gamma}\right) &\le \mathcal{O}\left(\left(\frac{\sqrt{\frac{\eta_x}{\eta_y}}L_R + \frac{2}{\eta_y}}{\frac{1}{\eta_y}}\right)\log\frac{1}{\gamma}\right) \\
&= \mathcal{O}\left(\left(\sqrt{\eta_x\eta_y}L_R + 1\right)\log\frac{1}{\gamma}\right)
\end{aligned}
$$

5 Conclusion

In this work, we investigated the composite saddle point problem $\min_x \max_y p(x) + R(x, y) - q(y)$, where $R(x, y)$ is L_R-smooth, μ_x-strongly convex, and μ_y-strongly concave, with convex and smooth functions $p(x)$ and $q(y)$. We introduced a new algorithm that achieves optimal overall complexity $\mathcal{O}\left(\left(\sqrt{\frac{L_p}{\mu_x}} + \frac{L_R}{\sqrt{\mu_x\mu_y}} + \sqrt{\frac{L_q}{\mu_y}}\right)\log\frac{1}{\varepsilon}\right)$, separating the oracle calls between the composite and saddle terms. Our method reduces the number of oracle calls to $\mathcal{O}\left(\left(\sqrt{\frac{L_p}{\mu_x}}+\sqrt{\frac{L_q}{\mu_y}}\right)\log\frac{1}{\varepsilon}\right)$ for $\nabla p(x)$ and $\nabla q(y)$, while requiring $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}},\sqrt{\frac{L_q}{\mu_y}},\frac{L_R}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$ for $\nabla R(x,y)$. To our knowledge, this is the first optimal algorithm that provides complexity separation in the case $\mu_x \neq \mu_y$. Moreover, we applied this algorithm to bilinear saddle point problems, achieving the best known complexity bounds for this class.

References

[ADS+19] Mohammad Alkousa, Darina Dvinskih, Fedor Stonyakin, Alexander Gasnikov, and Dmitry Kovalev. Accelerated methods for composite non-bilinear saddle point problem. *arXiv preprint arXiv: 1906.03620*, 2019.

[ASM+ar] Waïss Azizian, Damien Scieur, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. Accelerating smooth games by manipulating spectral shapes. *Unknown Journal*, Unknown Year.

[AT06] Alfred Auslender and Marc Teboulle. Interior gradient and proximal methods for convex and conic optimization. siam journal on optimization. *Unknown Journal*, 2006.

[BMR+20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell et al. Language models are few-shot learners. *Unknown Journal*, 2020.

[BSG20] Alexander Beznosikov, Valentin Samokhin, and Alexander Gasnikov. Distributed sadde-point problems: lower bounds and optimal and robust algorithms. *arXiv preprint arXiv:2010.13112*, 2020.

[CLO17] Yunmei Chen, Guanghui Lan, and Yuyuan Ouyang. Accelerated schemes for a class of variational inequalities. *Unknown Journal*, 2017.

[CP11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Unknown Journal*, 2011.

[CRG23] Savelii Chezhegov, Alexander Rogozin, and Alexander Gasnikov. On decentralized nonsmooth optimization. *arXiv preprint arXiv:2303.08045*, 2023.

[CST21] Michael B. Cohen, Aaren Sidfort, and Kevin Tian. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. *arXiv preprint arXiv: 2011.06572*, 2021.

[DCL+17] Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. *Unknown Journal*, 2017.

[DGJL22] Simon S. Du, Gauthier Gidel, Michael I. Jordan, and Chris Junchi Li. Optimal extragradient-based bilinearly-coupled saddle-point optimization. *arXiv preprint arXiv: 2206.08573*, 2022.

[GBV+18] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. *arXiv preprint arXiv:1802.10551*, 2018.

[GDG19] Eduard Gorbunov, Darina Dvinskikh, and Alexander Gasnikov. Optimal decentralized distributed algorithms for stochastic convex optimization. *arXiv preprint arXiv:1911.07363*, 2019.

[Gul91] Osman Guler. On the convergence of the proximal point algorithm for convex minimization. *Unknown Journal*, 1991.

[IAGMar] Adam Ibrahim, Waïss Azizian, Gauthier Gidel, and Ioannis Mitliagkas. Linear lower bounds and conditioning of differentiable games. *Unknown Journal*, Unknown Year.

[JS20] Yujia Jin and Aaron Sidford. Efficiently solving MDPs with stochastic mirror descent. *Unknown Journal*, 2020.

[JST22] Yujia Jin, Aaron Sidford, and Kevin Tian. Sharper rates for separable minimax and finite sum optimization via primal-dual extragradient methods. *Unknown Journal*, 2022.

[KG22] Dmitry Kovalev and Alexander Gasnikov. The first optimal algorithm for smooth and strongly-convex-strongly-concave minimax optimizatio. *Advances in Neural Information Processing Systems*, 2022.

[KGR22] Dmitry Kovalev, Alexander Gasnikov, and Peter Richtárik. Accelerated primal-dual gradient method for smooth and convex-concave saddle-point problems with bilinear coupling. *Unknown Journal*, 2022.

[Korar] G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Unknown Journal*, Unknown Year.

[KSR20] Dmitry Kovalev, Adil Salim, and Peter Richtarik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Unknown Journal*, 2020.

[Lan16] Guanghui Lan. Gradient sliding for composite optimization. *Unknown Journal*, 2016.

[LFYL20] Huan Li, Cong Fang, Wotao Yin, and Zhouchen Lin. Decentralized accelerated gradient methods with increasing penalty parameters. *Unknown Journal*, 2020.

[LJJ20] Tianyi Lin, Chi Jin, and Michael I. Jordan. Near-optimal algorithms for minimax optimization. *Unknown Journal*, 2020.

[LO21] Guanghui Lan and Yuyuan Ouyang. Mirror-prox sliding methods for solving a class of monotone variational inequalities. *arXiv preprint arXiv: 2111.00996*, 2021.

[LW16] Adrian S Lewis and Stephen J Wright. A proximal method for composite minimization. *Unknown Journal*, 2016.

[LYG+23] Chris Junchi LI, Angela Yuan, Gidel Gauthier, Gu Quanquan, and Michael Jordan. Stochastic variance reduction methods for policy evaluationnesterov meets optimism: Rate-optimal separable minimax optimization. *Unknown Journal*, 2023.

[MOP19] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. *Unknown Journal*, 2019.

[MS10] Renato D. C. Monteiro and B. F. Svaiter. Complexity of variants of tseng's modified f-b splitting and korpelevich's methods for generalized variational inequalities with applications to saddle point and convex optimization problems. *Unknown Journal*, 2010.

[Nes18] Yurii Nesterov. Lectures on convex optimization and volume 137. *Unknown Journal*, 2018.

[Nesar] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Unknown Journal*, Unknown Year.

[NSar] Yurii Nesterov and L. Scrimali. Solving strongly monotone variational and quasi-variational inequalities. *Unknown Journal*, Unknown Year.

[Roc76] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. siam journal on control and optimization. *Unknown Journal*, 1976.

[RYKG22] Alexander Rogozin, Demyan Yarmoshik, Ksenia Kopylova, and Alexander Gasnikov. Decentralized strongly-convex optimization with affine constraints: Primal and dual approaches. *Unknown Journal*, 2022.

[SBB+17] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. *Unknown Journal*, 2017.

[SCKRar] Adil Salim, Laurent Condat, Dmitry Kovalev, and Peter Richtarik. An optimal algorithm for strongly convex minimization under affine constraints. *Unknown Journal*, Unknown Year.

[SCST17] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017.

[SNRar] Stefano Savazzi, Monica Nicoli, and Vittorio Rampa. Federated learning with cooperating devices: A consensus approach for massive iot networks. *Unknown Journal*, Unknown Year.

[SSBDar] Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning - from theory to algorithms. *Unknown Journal*, Unknown Year.

[THOar] Kiran K. Thekumparampil, Niao He, and Sewoong Oh. Lifted primal-dual method for bilinearly coupled smooth minimax optimization. *Unknown Journal*, Unknown Year.

[Tse00] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *Unknown Journal*, 2000.

[Tse08] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

[WL20] Yuanhao Wang and Jian Li. Improved algorithms for convex-concave minimax optimization. *Unknown Journal*, 2020.

[WWKS18] Weiran Wang, Jialei Wang, Mladen Kolar, and Nathan Srebro. Distributed stochastic multi-task learning with graph regularization. *arXiv preprint arXiv:1802.03830*, 2018.

[XHZ21] Guangzeng Xie, Yuze Han, and Zhihua Zhang. Dippa: An improved method for bilinear saddle point problems. *arXiv preprint arXiv: 2103.08270*, 2021.

[XYLC19] Lin Xiao, Adams Wei Yu, Qihang Lin, and Weizhu Chen. Dscovr: Randomized primal-dual block coordinate algorithms for asynchronous distributed optimization. *Unknown Journal*, 2019.

[ZHZ19] Junyu Zhang, Mingyi Hong, and Shuzhong Zhang. On lower iteration complexity bounds for the saddle point problems. *arXiv preprint arXiv:1912.07481*, 2019.

In Appendix we provide discussions for various cases. Using similar reasonings as in §3.3 lower bounds can be obtained subsequently. See Table 1 for a complete view of iteration complexity separation.

A Convex-concave and strongly-convex-concave composite SPP

For the convex-concave composite SPP we assume that $\mu_x = \mu_y = 0$ that means $R(x, y)$ is convex-concave. For the strongly-convex-concave composite SPP we assume that $\mu_x > \mu_y = 0$ that means $R(x, y)$ is strongly-convex-concave. To present the results for these problems we make standard assumption that solution (x^*, y^*) is limited, i.e. $\|x^*\| \leq \mathcal{D}_x$, $\|y^*\| \leq \mathcal{D}_y$. We use this assumption and consider problem (1) with regularization terms. For strongly-convex-concave case we regularize function $q(y)$ and consider the problem

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} \left\{ p(x) + R(x, y) - q(y) - \frac{\varepsilon}{12\mathcal{D}_y^2}\|y\|^2 \right\} \tag{31}$$

instead of the problem (1). For the convex-concave case we also add the regularization terms for the functions $p(x)$ and $q(y)$ and consider the problem

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} \left\{ p(x) + \frac{\varepsilon}{16\mathcal{D}_x^2}\|x\|^2 + R(x, y) - q(y) - \frac{\varepsilon}{16\mathcal{D}_y^2}\|y\|^2 \right\} \tag{32}$$

instead of the problem (1). To demonstrate the equivalence of problems (31), (32) with regularisation terms to problem (1) we present the following lemma.

Lemma 4. *Consider problem* (1) *under Assumptions 1 to 3. If $\mu_x > 0$, $\mu_y = 0$ (strongly-convex-concave case) and $(\widehat{x}, \widehat{y})$ is an $\frac{2\varepsilon}{3}$-solution to the problem* (31) *or if $\mu_x = 0$, $\mu_y = 0$ (convex-concave case) and $(\widehat{x}, \widehat{y})$ is an $\frac{\varepsilon}{2}$-solution to the problem* (32) *with $\|x^*\| \leq \mathcal{D}_x$, $\|y^*\| \leq \mathcal{D}_y$. Then, $(\widehat{x}, \widehat{y})$ is an ε-solution to problem* (1).

Due to this lemma we need to find an $\frac{2\varepsilon}{3}$-solution to the problem (31) or an $\frac{\varepsilon}{2}$-solution to the problem (32). To find them we apply Algorithm 1 with composites $p(x)$, $q(y)$. By Theorem 3 Algorithm 1 requires $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\sqrt{\mu_x\varepsilon}}\mathcal{D}_y\right\}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla R(x, y)$ and $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla p(x)$, $\nabla q(y)$ to find an ε-solution to the problem (1) in the strongly-convex-concave case and $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right\}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla R(x, y)$ and $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla p(x)$, $\nabla q(y)$ to find an ε-solution to (1) in the convex-concave case.

A.1 More Discussion: Convex-concave and strongly-convex-concave case

For the convex-concave case Algorithm 1 requires

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right) \quad \text{calls of} \quad \nabla p(x), \nabla q(y)$$

and

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right\}\log^2\frac{1}{\varepsilon}\right) \quad \text{calls of} \quad \nabla R(x,y)$$

to find an ε-solution to (1). This result achieves the lower bounds on iteration complexity $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ up to logarithmic factors and generalizes results

$$\mathcal{O}\left(\sqrt{\frac{\max\{L_p, L_q\}}{\varepsilon}}\mathcal{D}\right) \quad \text{calls of} \quad \nabla p(x), \nabla q(y)$$

and

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{\max\{L_p, L_q\}}{\varepsilon}}\mathcal{D}, \frac{L_R}{\varepsilon}\mathcal{D}^2\right\}\right) \quad \text{calls of} \quad \nabla R(x,y)$$

from [LO21], where $\mathcal{D} = \max\{\mathcal{D}_x, \mathcal{D}_y\}$.

Analogously for the strongly-convex-concave case, we obtain results with regularization by changing μ_y on $\frac{\varepsilon}{\mathcal{D}_y^2}$ for complexities in §3.3. We omit the discussions.

A.2 Lower bounds

In the *convex-concave* case the lower bounds are $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla R(x,y)$ and $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla p(x), \nabla q(y)$.

For *strongly-convex-concave* problem (1): $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\sqrt{\mu_x\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla R(x,y)$ and $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of $\nabla p(x), \nabla q(y)$ to find an ε-solution.

B Strongly-convex-strongly-concave Bilinear Saddle Point Problems (SPP)

In this section and the following two, we presents our results for bilinear saddle point problems. The bilinear strongly-convex-strongly-concave problem has the following form

$$\min_{x\in\mathbb{R}^{d_x}}\max_{y\in\mathbb{R}^{d_y}} p(x) + x^\top By - q(y) \tag{33}$$

To this problem we assume that the following assumptions hold

Assumption 6. $p(x): \mathbb{R}^{d_x} \to \mathbb{R}$ *is* L_p*-smooth and* μ_p*-strongly convex function*

Assumption 7. $q(y): \mathbb{R}^{d_y} \to \mathbb{R}$ *is* L_q*-smooth and* μ_q*-strongly convex function*

Assumption 8. *Matrix* $B: \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ *is positive semi-definite.*

To apply Algorithm 1 to the problem (33) we reformulate it as a problem

$$\min_x \max_y \widetilde{p}(x) + \frac{\mu_p}{2}\|x\|^2 + x^\top By - \frac{\mu_q}{2}\|y\|^2 - \widetilde{q}(y)$$

with composites $\widetilde{p}(x) = p(x) - \frac{\mu_p}{2}\|x\|^2$, $\widetilde{q}(y) = q(y) - \frac{\mu_y}{2}\|y\|^2$.

Auxiliary subproblem complexity. At each iteration of Algorithm 1 we need to find a γ-solution to the problem

$$\min_x \max_y \langle \nabla \widetilde{p}(x_g^k), x\rangle + \frac{1}{2\eta_x}\|x - x^k\|^2 + \frac{\mu_p}{2}\|x\|^2 + x^\top By - \frac{\mu_q}{2}\|y\|^2 - \frac{1}{2\eta_y}\|y - y^k\|^2 - \langle \nabla \widetilde{q}(y_g^k), y\rangle \quad (34)$$

with γ defined in (23). The simplest way to solve this problem is reformulate it as a minimization problem in x using the first order optimal condition in y:

$$B^\top x - \mu_q y - \frac{1}{\eta_y}(y - y^k) - \nabla \widetilde{q}(y_g^k) = 0$$

$$y(x) = \frac{1}{\frac{1}{\eta_y} + \mu_q}\left(B^\top x - \nabla \widetilde{q}(y_g^k) + \frac{1}{\eta_y} y^k\right)$$

After reformulation we get the quadratic problem

$$\min_x \langle x, Ax\rangle + \langle b, x\rangle + c$$

with

$$\begin{aligned}
A &= \frac{1}{2}\left(\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right) I + BB^\top\right),\\
b &= \nabla \widetilde{p}(x_g^k) - \frac{1}{\eta_y}x^k + \left(1 - \frac{2\eta_y}{1 + \eta_y\mu_q}\right) B\left(\nabla \widetilde{q}(y_g^k) - \frac{1}{\eta_y} y^k\right),\\
c &= \frac{\eta_y}{2(1 + \eta_y\mu_q)}\|\nabla \widetilde{q}(y_g^k)\|^2 - \frac{1}{1 + \mu_q\eta_y}\langle \nabla \widetilde{q}(y_g^k), y^k\rangle + \frac{\mu_q(1 - \eta_y\mu_q)}{2(1 + \eta_y\mu_q)^2}\|y^k\|^2
\end{aligned}$$

This problem can be solved by Nesterov's Accelerated Gradient Descent that requires

$$\begin{aligned}
T &= \mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}}\log\frac{1}{\gamma}\right) = \mathcal{O}\left(\sqrt{\frac{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right) + \lambda_{\max}(BB^\top)}{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right) + \lambda_{\min}(BB^\top)}}\log\frac{1}{\gamma}\right)\\
&= \mathcal{O}\left(\min\left\{\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}, \sqrt{1 + \frac{\lambda_{\max}(BB^\top)}{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right)}}\right\}\log\frac{\sqrt{\lambda_{\max}(BB^\top)}}{\min\{\mu_p, \mu_q\}}\right)
\end{aligned}$$

iterations or calls of oracles $B/B^\top$ to find an ε-solution to (34).

Overall complexity. Next, we make some computations to get the overall complexity of Algorithm 1 for bilinear case

$$K \times T = \mathcal{O}\left(\left(1 + \sqrt{\frac{L_p}{\mu_x}} + \sqrt{\frac{L_q}{\mu_y}}\right)\log\frac{1}{\varepsilon}\right) \times \mathcal{O}\left(\min\{T_1, T_2\}\log\frac{\sqrt{\lambda_{\max}(BB^\top)}}{\min\{\mu_p, \mu_q\}}\right)$$

where $T_1 = \sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}$ and $T_2 = \sqrt{1 + \frac{\lambda_{\max}(BB^\top)}{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right)}}$. Next we compute $K \times T_1$ and $K \times T_2$ for case $\frac{L_p}{\mu_p} \geq \frac{L_q}{\mu_q}$.

$$K \times T_1 = \mathcal{O}\left(\sqrt{\frac{L_p}{\mu_p}}\log\frac{1}{\varepsilon} \times \sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\right) = \mathcal{O}\left(\sqrt{\frac{L_p}{\mu_p}}\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\log\frac{1}{\varepsilon}\right)$$

$$
\begin{aligned}
K \times T_2 &= \mathcal{O}\left(\sqrt{\frac{L_p}{\mu_p}} \log\frac{1}{\varepsilon} \times \sqrt{1 + \frac{\lambda_{\max}(BB^\top)}{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right)}}\right) \\
&= \mathcal{O}\left(\sqrt{\frac{L_p}{\mu_p}} \log\frac{1}{\varepsilon} \times \left(1 + \sqrt{\lambda_{\max}(BB^\top)\eta_x\eta_y}\right)\right) \\
&= \mathcal{O}\left(\left(\sqrt{\frac{L_p}{\mu_p}} + \sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_q}\eta_x}\right)\log\frac{1}{\varepsilon}\right) \\
&= \mathcal{O}\left(\left(\sqrt{\frac{L_p}{\mu_p}} + \sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\mu_q}}\right)\log\frac{1}{\varepsilon}\right)
\end{aligned}
$$

The case $\frac{L_q}{\mu_q} \geq \frac{L_p}{\mu_p}$ is done similarly. These calculations allow us to formulate the following theorem about oracle complexities of Algorithm 1 applied to the problem (33).

Theorem 4. *Consider Problem* (33) *under Assumptions 6 to 8. Then, to find an ε-solution, Algorithm 1 requires*

$$
\mathcal{O}\left(\max\left\{1, \sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}\right\}\log\frac{1}{\varepsilon}\right) \quad \textit{calls of} \quad \nabla p(x), \nabla q(y)
$$

and

$$
\mathcal{O}\left(\min\{K_1, K_2\}\log\frac{\sqrt{\lambda_{\max}(BB^\top)}}{\min\{\mu_p, \mu_q\}}\log\frac{1}{\varepsilon}\right) \quad \textit{calls of} \quad B \ \textit{or} \ B^\top
$$

where

$$
K_1 = \max\left\{\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}}, \sqrt{\frac{L_q\lambda_{\max}(BB^\top)}{\mu_q\lambda_{\min}(BB^\top)}}\right\}
$$

and

$$
K_2 = \max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\mu_q}}\right\}
$$

B.1 More Discussion: Bilinear strongly-convex-strongly-concave case

For the bilinear strongly-convex-strongly-concave case (33) Algorithm 1 requires

$$
\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}\right\}\log\frac{1}{\varepsilon}\right) \quad \text{oracle calls of} \quad \nabla p(x), \nabla q(y)
$$

and

$$
\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \frac{L_B}{\sqrt{\mu_p\mu_q}}\right\}\log\frac{L_B}{\min\{\mu_p, \mu_q\}}\log\frac{1}{\varepsilon}\right) \quad \text{oracle calls of} \quad \nabla R(x,y)
$$

to find an ε-solution to (33). Also, the iteration complexity of Algorithm 1 is $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \frac{L_B}{\sqrt{\mu_p\mu_q}}\right\}\log\frac{L_B}{\min\{\mu_p,\mu_q\}}\log\frac{1}{\varepsilon}\right)$. The same results $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \frac{L_B}{\sqrt{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$ on iteration complexity were proposed in works [KGR22], [THOar], [DGJL22] but the main benefit of our approach is complexity separation.

B.2 Lower bounds

For the *bilinear strongly-convex-strongly-concave* problem (33) the lower bound on iteration complexity $\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \frac{L_B}{\sqrt{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$ was also proposed in [ZHZ19]. Problem

$$\min_x \max_y p(x) + \sqrt{\mu_p\mu_q}\langle x, y\rangle - q(y)$$

is a special case of (33) with $L_B = \sqrt{\mu_p\mu_q}$ and $B = \sqrt{\mu_p\mu_q}I$. It means that the lower bound on oracle calls of $\nabla p(x), \nabla q(y)$ to problem (33) is $\Omega\left(\max\left\{1, \sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$. Besides, problem

$$\min_x \max_y \frac{\mu_p}{2}\|x\|^2 + x^\top By - \frac{\mu_q}{2}\|y\|^2$$

also a special case of (33) that requires $\Omega\left(\max\left\{1, \frac{L_B}{\sqrt{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$ oracle calls of B or $B^\top$ to find an ε-solution to this problem.

C Affinely constrained minimization

This problem has the following form:

$$\min_{\mathbf{B}x=c} p(x) \tag{35}$$

where $c \in \mathrm{range}\mathbf{B}$. Also, $p(x)$ is μ_p-strongly convex function and $\mathbf{B}$ is positive definite (i. e. $\lambda_{\min}(\mathbf{B}\mathbf{B}^\top) > 0$). This problem is equivalent to saddle point problem:

$$\min_x \max_y p(x) + x^\top \mathbf{B}y - y^\top c \tag{36}$$

To apply Algorithm 1 to this problem we make regularization and get the following problem

$$\min_x \max_y p(x) + x^\top \mathbf{B}y - y^\top c - \frac{\varepsilon}{16\mathcal{D}_y^2}\|y\|^2$$

By Lemma 4, if we find $\frac{2\varepsilon}{3}$-solution to this problem, then we find an ε-solution to (36). To find this solution we apply Algorithm 1 and get the following complexity.

Corollary 1. *Consider Problem* (36)*. Then, to find an ε-solution, Algorithm 1 requires*

$$\mathcal{O}\left(\max\left\{1, \sqrt{\frac{L_p}{\mu_p}}\right\}\log\frac{1}{\varepsilon}\right) \quad \textit{calls of} \;\; \nabla p(x)$$

and

$$\mathcal{O}\left(\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}}\log^2\frac{1}{\varepsilon}\right) \quad \textit{calls of} \;\; B \;\; \textit{or} \;\; B^\top$$

This corollary is derived from Theorem 4 and the fact that $\min\{a, b\} \le a$.

C.1 More Discussion: Affinely constrained minimization case

For the affinely constrained minimization case (36) Algorithm 1 requires

$$\mathcal{O}\left(\sqrt{\frac{L_p}{\mu_p}}\log\frac{1}{\varepsilon}\right) \quad \text{oracle calls of} \quad \nabla p(x)$$

and

$$\mathcal{O}\left(\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}}\log^2\frac{1}{\varepsilon}\right) \quad \text{calls of} \quad B \text{ or } \quad B^\top$$

This matches the iteration complexity of algorithms from the works [KSR20], [KGR22] up to logarithmic factor. Note, in these works, the authors achieve the lower bounds [SCKRar] exactly. But the key idea of Algorithm 1 in separating oracle complexities.

Meanwhile, we can apply this results to distributed optimization problem (4). For this problem Algorithm 1 requires $\mathcal{O}\left(\sqrt{\frac{L_F}{\mu_F}}\log\frac{1}{\varepsilon}\right)$ calls of $\nabla F(\mathbf{x})$, i.e. local oracle calls and $\mathcal{O}\left(\sqrt{\frac{L_F\lambda_{\max}(W)}{\mu_F\lambda^+_{\min}(W)}}\log^2\frac{1}{\varepsilon}\right)$ calls of W, i.e. communication rounds. Algorithm 1 achieves the lower bounds for distributed optimization [SBB+17] up to logarithmic factor. The optimal method for this problem was proposed in [BSG20].

C.2 Lower bounds

For *affinely constrained minimization* problem (36) the lower bound on oracle calls of $\nabla p(x)$ is $\Omega\left(\sqrt{\frac{L_p}{\mu_p}}\log\frac{1}{\varepsilon}\right)$ and the lower bound on calls of $B/B^\top$ is $\Omega\left(\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}}\log\frac{1}{\varepsilon}\right)$.

D Bilinear problem with linear composites

In this subsection we consider bilinear problem with linear composites:

$$\min_x \max_y x^\top d + x^\top \mathbf{B} y - y^\top c \tag{37}$$

where matrix $\mathbf{B}$ is positive definite ($\lambda_{\min}(BB^\top) = \lambda^+_{\min}(BB^\top)$). As in the previous subsection, we make the regularization to apply Algorithm 1. The problem (37) with regularization has the following form:

$$\min_x \max_y \frac{\varepsilon}{16\mathcal{D}_x^2}\|x\|^2 + x^\top d + x^\top \mathbf{B} y - y^\top c - \frac{\varepsilon}{16\mathcal{D}_y^2}\|y\|^2 \tag{38}$$

We need to find an $\frac{\varepsilon}{2}$-solution to find an ε-solution to (37) by Lemma 4. To find it we apply Algorithm 1 with the following complexity.

Corollary 2. *Consider Problem* (37)*. Then, to find an ε-solution, Algorithm 1 requires*

$$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\log^2\frac{1}{\varepsilon}\right) \quad \textit{calls of} \quad B \textit{ or } \quad B^\top$$

D.1 More Discussion: Billinear case with linear composites

For the billinear case with linear composites (37) Algorithm 1 requires

$$\mathcal{O}\left(\log\frac{1}{\varepsilon}\right) \text{ oracle calls of } \nabla p(x), \nabla q(y)$$

and

$$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\log^2\frac{1}{\varepsilon}\right) \text{ oracle calls of } B, B^\top$$

This results match the iteration complexity from the work [ASM$^+$ar] up to logarithmic factor. In contrast to our results, in work [ASM$^+$ar] the lower bounds [IAGMar] are achieved.

D.2 Lower bounds

For *billinear problem with linear composites* (37) the lower bound on oracle calls of $\nabla p(x), \nabla q(y)$ is $\Omega\left(\log\frac{1}{\varepsilon}\right)$ and the lower bound on calls of $B/B^\top$ is $\Omega\left(\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\log\frac{1}{\varepsilon}\right)$.

<table>
<tr><th></th><th>Reference</th><th>Oracle calls of $\nabla p(x), \nabla q(y)$</th><th>Oracle calls of $\nabla R(x,y)$ or $B, B^\top$</th><th>Compl. Sep.</th></tr>
<tr><td></td><td colspan="4">Strongly-convex-strongly-concave case</td></tr>
<tr><td rowspan="6">Upper</td><td>Korpelevich, 1974 [Korar]
Tseng, 2000 [Tse00]
Nesterov and Scimali, 2006 [NSar]
Gidel et al., 2018 [GBV+18]</td><td colspan="2">$\mathcal{O}\left(\left(\frac{L_R+L_p}{\mu_x}+\frac{L_R+L_q}{\mu_y}\right)\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Lin et al., 2020 [LJJ20]</td><td colspan="2">$\mathcal{O}\left(\frac{L_R+\sqrt{L_pL_q}}{\sqrt{\mu_x\mu_y}}\log^3\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Wang and Li, 2020 [WL20]</td><td colspan="2">$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}},\sqrt{\frac{L_q}{\mu_y}},\sqrt{\frac{L_RL}{\mu_x\mu_y}}\right\}\log^3\frac{(L_p+L_R)(L_q+L_R)}{\mu_x\mu_y}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Kovalev and Gasnikov, 2022 [KG22]</td><td colspan="2">$\mathcal{O}\left(\frac{L_R+\sqrt{L_pL_q}}{\sqrt{\mu_x\mu_y}}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Jin et al., 2022 [JST22]
Li et al., 2022 [LYG+23]</td><td colspan="2">$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}},\sqrt{\frac{L_q}{\mu_y}},\frac{L_R}{\mu_x},\frac{L_R}{\mu_y}\right\}\log\frac{1}{\varepsilon}\right)$[1]</td><td>✗</td></tr>
<tr><td>This paper</td><td>$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}},\sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}},\sqrt{\frac{L_q}{\mu_y}},\frac{L_R}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)$</td><td>✓</td></tr>
<tr><td rowspan="2">Lower</td><td>Zhang et al., 2019 [ZHZ19]</td><td>–</td><td>$\Omega\left(\frac{L_R}{\sqrt{\mu_x\mu_y}}\log\frac{1}{\varepsilon}\right)$</td><td>–</td></tr>
<tr><td>Nesterov, 2004 [Nes18]</td><td>$\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}},\sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>–</td><td>–</td></tr>
<tr><td></td><td colspan="4">Convex-concave case</td></tr>
<tr><td rowspan="4">Upper</td><td>Korpelevich, 1974 [Korar]
Tseng, 2000 [Tse00]
Monteiro and Svaiter, 2010 [MS10]</td><td colspan="2">$\mathcal{O}\left(\frac{L\mathcal{D}^2}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Chen et al., 2017 [CLO17]</td><td colspan="2">$\mathcal{O}\left(\sqrt{\frac{\max\{L_p,L_q\}}{\varepsilon}}\mathcal{D}+\frac{L_R}{\varepsilon}\mathcal{D}^2\right)$</td><td>✗</td></tr>
<tr><td>Lan and Ouyang, 2021 [LO21]</td><td>$\mathcal{O}\left(\sqrt{\frac{\max\{L_p,L_q\}}{\varepsilon}}\mathcal{D}\right)$</td><td>$\mathcal{O}\left(\max\left\{\sqrt{\frac{\max\{L_p,L_q\}}{\varepsilon}}\mathcal{D},\frac{L_R}{\varepsilon}\mathcal{D}^2\right\}\right)$</td><td>✓</td></tr>
<tr><td>This paper</td><td>$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x,\sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$</td><td>$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x,\sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y,\frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right\}\log^2\frac{1}{\varepsilon}\right)$</td><td>✓</td></tr>
<tr><td rowspan="2">Lower</td><td>Zhang et al., 2019 [ZHZ19]</td><td>–</td><td>$\Omega\left(\frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right)$</td><td>–</td></tr>
<tr><td>Nesterov, 2004 [Nes18]</td><td>$\Omega\left(\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x+\sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right)$</td><td>–</td><td>–</td></tr>
<tr><td></td><td colspan="4">Bilinear strongly-convex-strongly-concave case</td></tr>
<tr><td rowspan="6">Upper</td><td>Korpelevich, 1976 [Korar]
Nesterov and Scrimali, 2006 [NSar]
Mokhtari et al., 2020 [MOP19]</td><td colspan="2">$\mathcal{O}\left(\frac{L}{\min\{\mu_p,\mu_q\}}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Cohen et al., 2021 [CST21]</td><td colspan="2">$\mathcal{O}\left(\max\left\{\frac{L_p}{\mu_p},\frac{L_q}{\mu_q},\sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Wang and Li, 2020 [WL20]</td><td colspan="2">$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}},\sqrt{\frac{L_q}{\mu_q}},\sqrt{\frac{L\sqrt{\lambda_{\max}(BB^\top)}}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Xie et al., 2021 [XHZ21]</td><td colspan="2">$\mathcal{O}\left(\max\left\{\sqrt[4]{\frac{L_p^2L_q}{\mu_p^2\mu_q}},\sqrt[4]{\frac{L_q^2L_p}{\mu_q^2\mu_p}},\sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>Kovalev et al., 2021 [KGR22]
Thekumparampil et al., 2022 [THOar]
Jin et al., 2022 [JS20]
Du et al., 2022 [DGJL22]
Li et al., 2022 [LYG+23]</td><td colspan="2">$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}},\sqrt{\frac{L_q}{\mu_q}},\sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>This paper</td><td>$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}},\sqrt{\frac{L_q}{\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>$\mathcal{O}\left(\min\{K_1,K_2\}\log\frac{\sqrt{\lambda_{\max}(BB^\top)}}{\min\{\mu_p,\mu_q\}}\log\frac{1}{\varepsilon}\right)$
$K_1=\max\left\{\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}},\sqrt{\frac{L_q\lambda_{\max}(BB^\top)}{\mu_q\lambda_{\min}(BB^\top)}}\right\}$
$K_2=\max\left\{\sqrt{\frac{L_p}{\mu_p}},\sqrt{\frac{L_q}{\mu_q}},\sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\mu_q}}\right\}$</td><td>✓</td></tr>
<tr><td rowspan="2">Lower</td><td>Zhang et al., 2019 [ZHZ19]</td><td>–</td><td>$\Omega\left(\sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\mu_q}}\log\frac{1}{\varepsilon}\right)$</td><td>–</td></tr>
<tr><td>Nesterov, 2004 [Nes18]</td><td>$\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}},\sqrt{\frac{L_q}{\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$</td><td>–</td><td>–</td></tr>
<tr><td></td><td colspan="4">Affinely constrained minimization case</td></tr>
<tr><td rowspan="2">Upper</td><td>Kovalev et al., 2020 [KSR20]
Kovalev et al., 2021 [KGR22]</td><td colspan="2">$\mathcal{O}\left(\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>This paper</td><td>$\mathcal{O}\left(\sqrt{\frac{L_p}{\mu_p}}\log\frac{1}{\varepsilon}\right)$</td><td>$\mathcal{O}\left(\min\left\{\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}},\max\left\{\sqrt{\frac{L_p}{\mu_p}},\sqrt{\frac{\lambda_{\max}(BB^\top)}{\mu_p\varepsilon}}\mathcal{D}_y\right\}\right\}\log^2\frac{1}{\varepsilon}\right)$</td><td>✓</td></tr>
<tr><td rowspan="2">Lower</td><td>Salim et al., 2021 [SCKRar]</td><td>–</td><td>$\Omega\left(\sqrt{\frac{L_p\lambda_{\max}(BB^\top)}{\mu_p\lambda_{\min}(BB^\top)}}\log\frac{1}{\varepsilon}\right)$</td><td>–</td></tr>
<tr><td>Nesterov, 2004 [Nes18]</td><td>$\Omega\left(\sqrt{\frac{L_p}{\mu_p}}\log\frac{1}{\varepsilon}\right)$</td><td>–</td><td>–</td></tr>
<tr><td></td><td colspan="4">Billinear case with linear composites</td></tr>
<tr><td rowspan="2">Upper</td><td>Azizian et al., 2020 [ASM+ar]</td><td colspan="2">$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\log\frac{1}{\varepsilon}\right)$</td><td>✗</td></tr>
<tr><td>This paper</td><td>$\mathcal{O}\left(\log\frac{1}{\varepsilon}\right)$</td><td>$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\log^2\frac{1}{\varepsilon}\right)$</td><td>✓</td></tr>
<tr><td>Lower</td><td>Ibrahim et al., 2020 [IAGMar]</td><td>–</td><td>$\Omega\left(\sqrt{\frac{\lambda_{\max}(BB^\top)}{\lambda_{\min}(BB^\top)}}\log\frac{1}{\varepsilon}\right)$</td><td>–</td></tr>
</table>

Table 1. Comparison of our results for finding an ε-solution with other works. In the strongly-convex-strongly-concave case, convergence is measured by the distance to the solution. In the convex-concave case, convergence is measured in terms of the gap function.