NEAR-OPTIMAL ALGORITHM WITH COMPLEXITY SEPARATION FOR STRONGLY CONVEX-STRONGLY CONCAVE COMPOSITE SADDLE POINT PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we revisit the saddle-point problem $\min_x \max_y p(x)+R(x,y)-q(y)$, where the function $R(x,y)$ is L_R-smooth, μ_x-strongly convex, and μ_y-strongly concave, and the functions $p(x), q(y)$ are convex and L_p, L_q-smooth, respectively. We develop a new algorithm that achieves separation of complexities with respect to the computation of the gradients $\nabla R(x,y)$ and $\nabla p(x)$, $\nabla q(y)$. In particular, our algorithm requires $\mathcal{O}\left(\left(\sqrt{\frac{L_p}{\mu_x}}+\sqrt{\frac{L_q}{\mu_y}}+\frac{L_R}{\sqrt{\mu_x\mu_y}}\right)\log\frac{1}{\varepsilon}\right)$ computations of the gradient $\nabla R(x,y)$ and $\mathcal{O}\left(\left(\sqrt{\frac{L_p}{\mu_x}}+\sqrt{\frac{L_q}{\mu_y}}\right)\log\frac{1}{\varepsilon}\right)$ computations of the gradients $\nabla p(x)$, $\nabla q(y)$ to find an ϵ-accurate solution to the problem. Moreover, under the condition $L_R \geq \sqrt{\mu_x L_q + \mu_y L_p}$, the algorithm becomes optimal (up to logarithmic factors), i.e., it cannot be improved due to the existing lower complexity bounds. To the best of our knowledge, our algorithm is the first to achieve near-optimal complexity separation in the case when $\mu_x \neq \mu_y$.

1 INTRODUCTION

In this paper, we consider the following composite saddle-point problem:

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

where $p(x)\colon \mathbb{R}^{d_x} \to \mathbb{R}$, $q(y)\colon \mathbb{R}^{d_y} \to \mathbb{R}$ and $R(x,y)\colon \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ are smooth functions. Problems of the form (1) have been actively studied in economics, game theory, statistics, and computer science (Başar & Olsder, 1998; Roughgarden, 2010; Von Neumann & Morgenstern, 1947; Facchinei & Pang, 2007; Berger, 2013). In addition, many applications of such problems have recently appeared in machine learning, including prediction and regression problems (Taskar et al., 2005; Xu et al., 2009), reinforcement learning (Du et al., 2017; Dai et al., 2018), adversarial training (Madry et al., 2017; Sinha et al., 2017), and generative adversarial networks (Goodfellow et al., 2014; Arjovsky et al., 2017).

1.1 CONVEX-CONCAVE SETTING AND SEPARATION OF COMPLEXITIES

In this work, we are interested in solving problem (1) in the fundamental strongly-convex-strongly-concave setting. In particular, we assume that the function $R(x,y)$ is strongly convex in the variable x and strongly concave in the variable y, while the regularization functions $p(x)$ and $q(y)$ are convex. Under these assumptions, problem (1) has a unique solution. Although saddle-point problems of this type have been studied for many years in the optimization literature, it turns out that there are very important questions related to these problems that have yet to be answered.

The natural goal of our paper is to provide an efficient algorithm for solving problem (1). In particular, we aim to develop an algorithm that can find an approximate solution to the problem using the fewest possible number of computations of the gradients $\nabla R(x,y)$ and $\nabla p(x)$, $\nabla q(y)$. The existing results of Nesterov (2018); Zhang et al. (2019) suggest that to find an ε-accurate solution (see Definition 3) to problem (1), any first-order optimization algorithm requires at least the following number of

computations of the gradient $\nabla R(x, y)$:

$$\Omega\left(\frac{L_R}{\sqrt{\mu_x \mu_y}} \log \frac{1}{\varepsilon}\right), \tag{2}$$

and at least the following number of computations of the gradients $\nabla p(x)$ and $\nabla q(y)$:

$$\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}\right\} \log \frac{1}{\varepsilon}\right), \tag{3}$$

where μ_x and μ_y are the strong convexity constants (see Definition 2) associated with the functions $p(x)$ and $q(y)$, respectively, and L_p, L_q, and L_R are the smoothness constants (see Definition 1) associated with the functions $p(x)$, $q(y)$, and $R(x, y)$, respectively. Unfortunately, the existing state-of-the-art algorithms (Kovalev & Gasnikov, 2022; Jin et al., 2022; Li et al., 2023) are unable to reach these lower complexity bounds. The algorithm of Kovalev & Gasnikov (2022) requires

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p L_q}{\mu_x \mu_y}}, \frac{L_R}{\sqrt{\mu_x \mu_y}}\right\} \log \frac{1}{\varepsilon}\right) \tag{4}$$

and the algorithms of Jin et al. (2022); Li et al. (2023) require

$$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \frac{L_R}{\mu_x}, \frac{L_R}{\mu_y}\right\} \log \frac{1}{\varepsilon}\right) \tag{5}$$

computations of the gradients of both $\nabla R(x, y)$ and $\nabla p(x)$, $\nabla q(y)$. One can observe that there is a significant gap between these upper bounds and the lower bounds (2) and (3).

The key problem with the existing state-of-the-art algorithms is that they have the same gradient computation complexity for both $\nabla R(x, y)$ and $\nabla p(x)$, $\nabla q(y)$. However, lower bounds on these computation complexities (eqs. (2) and (3), respectively) can be largely unbalanced, for instance, when $L_R/\sqrt{\mu_x \mu_y} \gg \sqrt{L_p/\mu_x + L_q/\mu_y}$. Hence, to reach (or at least come closer to) both lower complexity bounds (2) and (3), an efficient iterative algorithm for solving problem (1) would need to skip evaluations of $\nabla R(x, y)$ or $\nabla p(x)$, $\nabla q(y)$ from time to time. This would help to achieve different gradient computation complexities for $\nabla R(x, y)$ and $\nabla p(x)$, $\nabla q(y)$, which is called *complexity separation*. Thus, we arrive at the crucial task of developing an efficient algorithm with complexity separation for solving problem (1) which is able to reach one of the lower complexity bounds (2) or (3), or, under some circumstances, even both of them. Further, we provide an additional motivation for this task.

1.2 On the Importance of Complexity Separation

The development of efficient algorithms with complexity separation for solving problem (1) is an important challenge. It is motivated by theoretical interest in various aspects of optimization techniques, as well as numerous applications in practice. Further, we provide a few examples of such applications.

Distributed optimization. One of the common applications of problem (1) is decentralized distributed optimization. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, which represents a communication network of n compute nodes $i \in \mathcal{V} = \{1, \ldots, n\}$. The nodes can communicate across the communication links $e \in \mathcal{E}$ in the network and aim to solve the following finite-sum minimization problem:

$$\min_{x \in \mathbb{R}^d} \textstyle\sum_{i \in \mathcal{V}} f_i(x), \tag{6}$$

where each function $f_i(x) \colon \mathbb{R}^d \to \mathbb{R}$ is stored locally by the corresponding node $i \in \mathcal{V}$. Problem (6) is often reformulated as the following saddle-point problem (Kovalev et al., 2020):

$$\min_{x \in \mathbb{R}^{nd}} F(x) + \langle x, (\mathbf{W} \otimes \mathbf{I}_d) y \rangle, \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the so-called gossip matrix associated with the network $\mathcal{G}$, and $F(x) \colon \mathbb{R}^{nd} \to \mathbb{R}$ is a block-separable function, which is defined as $F(x_1, \ldots, x_n) = \sum_{i=1}^n f_i(x_i)$. It is easy to observe that reformulation (7) is indeed a special case of problem (1) with $R(x, y) = \langle x, (\mathbf{W} \otimes \mathbf{I}_d) y \rangle$ and $p(x) = F(x)$, $q(y) = 0$. Moreover, one can observe that computing $\nabla p(x)$ corresponds to the local computation of the gradients of the objective functions $f_i(x)$, while computing $\nabla R(x, y)$ corresponds to decentralized communication in the network. In this setting, the complexity separation is a highly

desired property. Indeed, in some cases, the communication network can be very slow, thus we would like to save communication rounds by performing multiple local gradient computations between them (Savazzi et al., 2020; Brown et al., 2020). On the other hand, the opposite scenario is also possible, where we want to save local gradient computations instead. For instance, this could occur during the training of a large machine learning model on a cluster of compute nodes connected over a fast communication network.

Personalized federated learning. Another important special case of problem (1) is the following personalized federated saddle-point problem (Smith et al., 2017; Wang et al., 2018; Li et al., 2020; Gorbunov et al., 2019):

$$\min_{x\in\mathbb{R}^{nd_x}}\max_{y\in\mathbb{R}^{nd_y}} \tfrac{\lambda}{2}\|x\|^2_{\mathbf{W}\otimes\mathbf{I}_{d_x}} + \textstyle\sum_{i\in\mathcal{V}} f_i(x_i,y_i) - \tfrac{\lambda}{2}\|y\|^2_{\mathbf{W}\otimes\mathbf{I}_{d_y}}, \tag{8}$$

where we use the notation $x = (x_1,\ldots,x_n)$ and $y = (y_1,\ldots,y_n)$, (x_i,y_i) corresponds to the weights of the local model stored by the node $i \in \mathcal{V}$, and $f_i(x_i,y_i)\colon \mathbb{R}^{d_x}\times\mathbb{R}^{d_y}\to\mathbb{R}$ are the objective functions. Similarly to the previous example, the complexity separation is desired to efficiently balance the costs of communication and local computation.

Empirical risk minimization. Finally, one of the most popular examples of problem (1) is the classical Empirical Risk Minimization problem (Shalev-Shwartz & Ben-David, 2014)

$$\min_{x\in\mathbb{R}^{d_x}}\left[p(x)+\ell(\mathbf{B}^\top x)\right] = \min_{x\in\mathbb{R}^{d_x}}\max_{y\in\mathbb{R}^{d_y}}\left[p(x)+\langle x,\mathbf{B}y\rangle - \ell^*(y)\right], \tag{9}$$

where $\ell(y)\colon \mathbb{R}^{d_y}\to\mathbb{R}$ is some convex loss function, $\ell^*(y)\colon \mathbb{R}^{d_y}\to\mathbb{R}$ is its Fenchel conjugate, $p(x)$ is a regularizer, and $\mathbf{B}\in\mathbb{R}^{d_x\times d_y}$ is the feature matrix. Once again, the desire for complexity separation could be motivated by reducing communication costs (Xiao et al., 2019), or computation costs, for instance when the gradients $\nabla p(x)$, $\nabla\ell(y)$, or $\nabla\ell^*(y)$ are hard to compute.

1.3 Main Contributions

In this paper, we resolve the task of finding an efficient algorithm with complexity separation for solving problem (1). In particular, we provide the following main contributions:

- We develop a new state-of-the-art Algorithm 1. This algorithm achieves (up to a logarithmic factor) the lower gradient computation complexity bound (3) for the gradients $\nabla p(x)$, $\nabla q(y)$, while improving the state-of-the-art gradient computation complexities (4) and (5) of Kovalev & Gasnikov (2022); Jin et al. (2022); Li et al. (2023) for the gradient $\nabla R(x,y)$. Moreover, under the condition $L_R \geq \sqrt{\mu_x L_q + \mu_y L_p}$, the proposed algorithm becomes *near-optimal*, that is, it achieves both lower bounds (2) and (3) (up to logarithmic factors).
- We use our approach and the proposed Algorithm 1 to obtain state-of-the-art complexity results for solving saddle-point problems with a bilinear coupling function in the Appendix.

2 Notation and Assumptions

In this paper, we use the following notation: $\mathbf{I}_d$ denotes the $d\times d$ identity matrix, $\|\cdot\|$ denotes the standard Euclidean norm, $\langle\cdot,\cdot\rangle$ denotes the standard scalar product. For vectors $x,x'\in\mathbb{R}^d$ and a symmetric positive definite matrix $\mathbf{A}\in\mathbb{R}^{d\times d}$, $\langle x,x'\rangle_{\mathbf{A}} = \langle x,\mathbf{A}x'\rangle$ denotes the weighted scalar product, and $\|x\|_{\mathbf{A}} = \sqrt{\langle x,x\rangle_{\mathbf{A}}}$ denotes the weighted Euclidean norm. For a differentiable function $f(x)\colon \mathbb{R}^d\to\mathbb{R}$, by $\mathrm{D}_f(x_1,x_2)\colon \mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ we denote the Bregman divergence associated with the function $f(x)$, which is defined as follows:

$$\mathrm{D}_f(x_1,x_2) = f(x_1) - f(x_2) - \langle\nabla f(x_2), x_1 - x_2\rangle. \tag{10}$$

For a proper lower semi-continuous function $f(x)\colon \mathbb{R}^d\to\mathbb{R}$ and $\eta>0$, by $\mathrm{prox}_{\eta f}(x)\colon \mathbb{R}^d\to\mathbb{R}^d$ we denote the proximal operator which is defined as follows:

$$\mathrm{prox}_{\eta f}(x) = \arg\min_{x'\in\mathbb{R}^d}\left[f(x') + \tfrac{1}{2\eta}\|x'-x\|^2\right]. \tag{11}$$

We say that the function $f(x)$ is proximal-friendly if the proximal operator $\mathrm{prox}_{\eta f}(x)$ can be computed efficiently. As mentioned in Section 1, we assume that the saddle-point problem (1) is smooth and strongly-convex-strongly-concave. We formalize this through the following definitions and assumptions.

Definition 1. *A differentiable function $f(z)\colon \mathbb{R}^d \to \mathbb{R}$ is called L-smooth for $L \geq 0$, if its gradient is L-Lipschitz continuous. That is, for all $z_1, z_2 \in \mathbb{R}^d$, the following inequality holds:*

$$\|\nabla f(z_1) - \nabla f(z_2)\| \leq L\|z_1 - z_2\|. \tag{12}$$

Definition 2. *A differentiable function $f(z)\colon \mathbb{R}^d \to \mathbb{R}$ is called μ-strongly convex for $\mu \geq 0$ if, for all $z_1, z_2 \in \mathbb{R}^d$, the following inequality holds:*

$$\mathrm{D}_f(z_1, z_2) \geq \tfrac{\mu}{2}\|z_1 - z_2\|^2. \tag{13}$$

When $\mu = 0$, we say that $f(x)$ is a convex function.

Assumption 1. *Function $p(x)\colon \mathbb{R}^{d_x} \to \mathbb{R}$ is differentiable, L_p-smooth and convex.*

Assumption 2. *Function $q(y)\colon \mathbb{R}^{d_y} \to \mathbb{R}$ is differentiable, L_q-smooth and convex.*

Assumption 3. *Function $R(x, y)\colon \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ is differentiable and L_R-smooth. Moreover, for all $x \in \mathbb{R}^{d_x}$ and $y \in \mathbb{R}^{d_y}$, the function $R(\cdot, y)\colon \mathbb{R}^{d_x} \to \mathbb{R}$ is μ_x-strongly convex and the function $R(x, \cdot)\colon \mathbb{R}^{d_y} \to \mathbb{R}$ is μ_y-strongly concave, where $\mu_x \leq L_p$, $\mu_y \leq L_q$ and $\sqrt{\mu_x \mu_y} \leq L_R$.*

Finally, to specify the quality of an arbitrary approximate solution to problem (1), we use the following definition.

Definition 3. *A vector $(x, y) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ is called an ϵ-accurate solution to problem* (1) *if the following inequality holds:*

$$\|x - x^*\|^2 + \|y - y^*\|^2 \leq \varepsilon, \tag{14}$$

where (x^*, y^*) is the unique solution to problem (1)*.*

3 Optimal Algorithm

In this section, we present the key idea used to develop Algorithm 1. Subsequently, we provide the convergence theorem (Theorem 1) and the complexity theorem (Theorem 3) of Algorithm 1. Lastly, we express these complexities for problem (1), specifically when $p(x) = 0$ or $q(y) = 0$.

3.1 Idea

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

$$\min_x u(x) + v(x). \tag{15}$$

For simplicity, we assume that function $u(x) + v(x)$ is μ-strongly convex and functions $u(x), v(x)$ are L_u, L_v-smooth, respectively. A fundamental and intuitive approach to solve this problem is to apply Nesterov's Accelerated Gradient Descent Nesterov (1983), Nesterov (2018):

$$\begin{aligned} x_g^k &= \alpha x^k + (1-\alpha)x_f^k \\ x^{k+1} &= x^k - \eta(\nabla u(x_g^k) + \nabla v(x_g^k)) \\ x_f^{k+1} &= x_g^k + \alpha(x^{k+1} - x^k). \end{aligned} \tag{16}$$

This method requires $\mathcal{O}\left(\sqrt{\frac{L_u+L_v}{\mu}}\log\frac{1}{\varepsilon}\right)$ computations of the gradients of both $\nabla u(x)$ and $\nabla v(x)$ to find an ε-solution to (15). This method uses the fewest possible number of computations for both gradients: $\nabla u(x)$ and $\nabla v(x)$. However, it does not allow separation of the number of gradient computations for each function. This lack of separation could be significant in certain scenarios, as previously discussed. One of the ideas for obtaining a complexity-separated algorithm is to use proximal point methods. These methods use a proximal step instead of a gradient one. For example, the Accelerated Proximal Point Algorithm Rockafellar (1976), Guler (1991), Auslender & Teboulle (2006), Tseng (2008), Lewis & Wright (2016):

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

	References	Oracle calls of $\nabla p(x), \nabla q(y)$	Oracle calls of $\nabla R(x,y)$ or $\mathbf{B}, \mathbf{B}^T$	Compl. Sep.
	Strongly convex-strongly concave case			
Upper	Korpelevich (1976) Tseng (2000) Nesterov & Scrimali (2006) Gidel et al. (2018)	$\mathcal{O}\left(\left(\frac{L_R+L_p}{\mu_x}+\frac{L_R+L_q}{\mu_y}\right)\log\frac{1}{\varepsilon}\right)$		✗
Upper	Alkousa et al. (2019)	$\nabla p(x): \mathcal{O}\left(\sqrt{\frac{L_p}{\mu_x}}\log\frac{1}{\varepsilon}\right), \nabla q(y): \mathcal{O}\left(\frac{L_R}{\sqrt{\mu_x\mu_y}}\sqrt{\frac{L_q}{\mu_y}}\log^3\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{L_R\sqrt{L_R}}{\mu_y\sqrt{\mu_x}}\log^3\frac{1}{\varepsilon}\right)$	✓
Upper	Lin et al. (2020)	$\mathcal{O}\left(\frac{L_R+\sqrt{L_pL_q}}{\sqrt{\mu_x\mu_y}}\log^3\frac{1}{\varepsilon}\right)$		✗
Upper	Wang & Li (2020)	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \sqrt{\frac{L_RL}{\mu_x\mu_y}}\right\}\log^3\frac{(L_p+L_R)(L_q+L_R)}{\mu_x\mu_y}\log\frac{1}{\varepsilon}\right)$		✗
Upper	Kovalev & Gasnikov (2022)	$\mathcal{O}\left(\frac{L_R+\sqrt{L_pL_q}}{\sqrt{\mu_x\mu_y}}\log\frac{1}{\varepsilon}\right)$		✗
Upper	Jin et al. (2022) Li et al. (2023)	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \frac{L_R}{\mu_x}, \frac{L_R}{\mu_y}\right\}\log\frac{1}{\varepsilon}\right)$[1]		✗
Upper	**This paper**	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}, \frac{L_R}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)$	✓
Lower	Zhang et al. (2019)	–	$\Omega\left(\frac{L_R}{\sqrt{\mu_x\mu_y}}\log\frac{1}{\varepsilon}\right)$	–
Lower	Nesterov (2018)	$\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\mu_y}}\right\}\log\frac{1}{\varepsilon}\right)$	–	–
	Convex-concave case			
Upper	Korpelevich (1976) Tseng (2000) Monteiro & Svaiter (2010)	$\mathcal{O}\left(\frac{L\mathcal{D}^2}{\varepsilon}\right)$		✗
Upper	Chen et al. (2017)	$\mathcal{O}\left(\sqrt{\frac{\max\{L_p,L_q\}}{\varepsilon}}\mathcal{D}+\frac{L_R}{\varepsilon}\mathcal{D}^2\right)$		✗
Upper	Lan & Ouyang (2021)	$\mathcal{O}\left(\sqrt{\frac{\max\{L_p,L_q\}}{\varepsilon}}\mathcal{D}\right)$	$\mathcal{O}\left(\max\left\{\sqrt{\frac{\max\{L_p,L_q\}}{\varepsilon}}\mathcal{D}, \frac{L_R}{\varepsilon}\mathcal{D}^2\right\}\right)$	✓
Upper	**This paper**	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right\}\log^2\frac{1}{\varepsilon}\right)$	✓
Lower	Zhang et al. (2019)	–	$\Omega\left(\frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right)$	–
Lower	Nesterov (2018)	$\Omega\left(\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x+\sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right)$	–	–
	Bilinear strongly convex-strongly concave case			
Upper	Korpelevich (1976) Nesterov & Scrimali (2006) Mokhtari et al. (2019)	$\mathcal{O}\left(\frac{L}{\min\{\mu_p,\mu_q\}}\log\frac{1}{\varepsilon}\right)$		✗
Upper	Cohen et al. (2021)	$\mathcal{O}\left(\max\left\{\frac{L_p}{\mu_p}, \frac{L_q}{\mu_q}, \sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$		✗
Upper	Wang & Li (2020)	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \sqrt{\frac{L\sqrt{\lambda_{\max}(\mathbf{BB}^T)}}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$		✗
Upper	Xie et al. (2021)	$\mathcal{O}\left(\max\left\{\sqrt[4]{\frac{L_p^2L_q}{\mu_p^2\mu_q}}, \sqrt[4]{\frac{L_q^2L_p}{\mu_q^2\mu_p}}, \sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$		✗
Upper	Kovalev et al. (2022) Thekumparampil et al. (2022) Jin & Sidford (2020) Du et al. (2022) Li et al. (2023)	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$		✗
Upper	**This paper**	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\min\{K_1,K_2\}\log\frac{\sqrt{\lambda_{\max}(\mathbf{BB}^T)}}{\min\{\mu_p,\mu_q\}}\log\frac{1}{\varepsilon}\right)$ $K_1=\max\left\{\sqrt{\frac{L_p\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\lambda_{\min}(\mathbf{BB}^T)}}, \sqrt{\frac{L_q\lambda_{\max}(\mathbf{BB}^T)}{\mu_q\lambda_{\min}(\mathbf{BB}^T)}}\right\}$ $K_2=\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\mu_q}}\right\}$	✓
Lower	Zhang et al. (2019)	–	$\Omega\left(\sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\mu_q}}\log\frac{1}{\varepsilon}\right)$	–
Lower	Nesterov (2018)	$\Omega\left(\max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}\right\}\log\frac{1}{\varepsilon}\right)$	–	–
	Affinely constrained minimization case			
Upper	Kovalev et al. (2020) Kovalev et al. (2022)	$\mathcal{O}\left(\sqrt{\frac{L_p\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\lambda_{\min}(\mathbf{BB}^T)}}\log\frac{1}{\varepsilon}\right)$		✗
Upper	**This paper**	$\mathcal{O}\left(\sqrt{\frac{L_p}{\mu_p}}\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\min\left\{\sqrt{\frac{L_p\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\lambda_{\min}(\mathbf{BB}^T)}}, \max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\varepsilon}}\mathcal{D}_y\right\}\right\}\log^2\frac{1}{\varepsilon}\right)$	✓
Lower	Salim et al. (2022)	–	$\Omega\left(\sqrt{\frac{L_p\lambda_{\max}(\mathbf{BB}^T)}{\mu_p\lambda_{\min}(\mathbf{BB}^T)}}\log\frac{1}{\varepsilon}\right)$	–
Lower	Nesterov (2018)	$\Omega\left(\sqrt{\frac{L_p}{\mu_p}}\log\frac{1}{\varepsilon}\right)$	–	–
	Bilinear case with linear composites			
Upper	Azizian et al. (2020)	$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\lambda_{\min}(\mathbf{BB}^T)}}\log\frac{1}{\varepsilon}\right)$		✗
Upper	**This paper**	$\mathcal{O}\left(\log\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\lambda_{\min}(\mathbf{BB}^T)}}\log^2\frac{1}{\varepsilon}\right)$	✓
Lower	Ibrahim et al. (2020)	–	$\Omega\left(\sqrt{\frac{\lambda_{\max}(\mathbf{BB}^T)}{\lambda_{\min}(\mathbf{BB}^T)}}\log\frac{1}{\varepsilon}\right)$	–

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. *Notation:* 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 $\|x^*\| \leq \mathcal{D}_x$, $\|y^*\| \leq \mathcal{D}_y, \mathcal{D}=\max\{\mathcal{D}_x, \mathcal{D}_y\}$. For *bilinear case* $\lambda_{\max}(\mathbf{BB}^T) > 0$ and $\lambda_{\min}(\mathbf{BB}^T) \geq 0$ are maximum and minimum eigenvalue of $\mathbf{BB}^T$ respectively, $L=\max(L_p, L_q, \sqrt{\lambda_{\max}(\mathbf{BB}^T)})$, μ_p, μ_q - strong convexity and strong concavity constants of $p(x), q(y)$ respectively.

[1] In these papers, results close to the lower bounds were obtained but a slightly different notation was used. For more details see Section 4.1.

which uses a proximal step in function $v(x)$ and a gradient one in function $u(x)$. The stumbling block of this method is the computation of the proximal operator for the function $v(x)$, which can be computed precisely only for proximal-friendly functions. To adapt Accelerated Proximal

Point Algorithm for L_v-smooth and non-proximal friendly function $v(x)$, the proximal operator can be computed inexactly. Firstly, to present this idea in detail, we rewrite the line $x^{k+1} = \text{prox}_{\eta v}(x^k - \eta \nabla u(x_g^k))$, using (11) and the first optimal condition for the differentiable function $v(x)$, in implicit form:

$$x^{k+1} = x^k - \eta \left(\nabla v(x^{k+1}) + \nabla u(x_g^k) \right). \tag{17}$$

Usually, the addition sequence $\hat{x}^{k+1}$ is used to compute x^{k+1} for non-proximal friendly function. This sequence updates as $\hat{x}^{k+1} \approx \text{prox}_{\eta v}(x^k - \eta \nabla u(x_g^k))$ and stores the approximated value of the proximal operator, which can be found by solving problem (11) with algorithm $\mathcal{A}$. After that, the sequence x^{k+1} is updated by (17) using $\hat{x}^{k+1}$. Summing up the above, we get the following method, which is based on the sliding technique Lan (2016):

$$\begin{aligned} x_g^k &= \alpha x^k + (1-\alpha) x_f^k \\ \hat{x}^{k+1} &\approx \arg\min_x \left\{ v(x) + \langle \nabla u(x_g^k), x \rangle + \frac{1}{2\eta} \|x - x^k\|^2 \right\} \\ x^{k+1} &= x^k - \eta (\nabla u(x_g^k) + \nabla v(\hat{x}^{k+1})) \\ x_f^{k+1} &= x_g^k + \alpha(\hat{x}^{k+1} - x^k). \end{aligned} \tag{18}$$

If Nesterov's Accelerated Gradient Descent Nesterov (1983), Nesterov (2018) is used as algorithm $\mathcal{A}$, Accelerated Proximal Point Algorithm requires $\mathcal{O}\left(\sqrt{\frac{L_u}{\mu}} \log \frac{1}{\varepsilon}\right)$ computations of the gradient of $\nabla u(x)$ and $\mathcal{O}\left(\sqrt{\frac{L_v}{\mu}} \log \frac{1}{\varepsilon}\right)$ computations of the gradient of $\nabla v(x)$ to find an ε-solution. The application of the proximal method to the composite optimization problem (15) enables us to achieve complexity separation. Importantly, this is accomplished without compromising the method's optimality. We are now prepared to revisit the composite saddle point problem (1). It should be noted that for problem (15), our objective is to find a point $\hat{x}$ that satisfies the equation $\nabla u(\hat{x}) + \nabla v(\hat{x}) = 0$. Note that for problem (15), it is enough to find a point $\hat{x}$ such that $\nabla u(\hat{x}) + \nabla v(\hat{x}) = 0$. Similarly, for problem (1), our goal is to find a point $(\hat{x}, \hat{y})$ such that $\begin{pmatrix} \nabla p(\hat{x}) + \nabla_x R(\hat{x}, \hat{y}) \\ \nabla q(\hat{y}) - \nabla_y R(\hat{x}, \hat{y}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. This fact allows us to adapt approach (18) to problem (1) by replacing $\nabla u(x)$ with operator $A = \begin{pmatrix} \nabla p(x) \\ \nabla q(y) \end{pmatrix}$ and $\nabla v(x)$ with operator $B = \begin{pmatrix} \nabla_x R(x,y) \\ -\nabla_y R(x,y) \end{pmatrix}$, and develop Algorithm 1.

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**
  $x_g^k = \alpha x^k + (1-\alpha) x_f^k, \quad y_g^k = \alpha y^k + (1-\alpha) y_f^k$
  $(\hat{x}^{k+1}, \hat{y}^{k+1}) \approx \arg\min_{x \in \mathbb{R}^{d_x}} \arg\max_{y \in \mathbb{R}^{d_y}} A_\eta^k(x,y)$

$$A_\eta^k(x,y) := \langle \nabla p(x_g^k), x \rangle + \frac{1}{2\eta_x} \|x - x^k\|^2 + R(x,y) - \langle \nabla q(y_g^k), y \rangle - \frac{1}{2\eta_y} \|y - y^k\|^2 \tag{19}$$

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

3.2 Convergence of Algorithm 1

Now we are ready to present the linear convergence of Algorithm 1 when applied to problem 1 under Assumptions 1-3.

Theorem 1. *Consider the application of Algorithm 1 to solve problem 1 under Assumptions 1-3, with one of the following tuning*

$$1.\ \textit{If}\ \frac{L_p}{\mu_x} \geq \frac{L_q}{\mu_y} : \alpha = \min\left\{1, \sqrt{\frac{\mu_x}{L_p}}\right\}, \quad \eta_x = \min\left\{\frac{1}{3\mu_x}, \frac{1}{3L_p\alpha}\right\}, \quad \eta_y = \frac{\mu_x}{\mu_y}\eta_x; \tag{20}$$

$$2.\ \textit{If}\ \frac{L_q}{\mu_y} > \frac{L_p}{\mu_x} : \alpha = \min\left\{1, \sqrt{\frac{\mu_y}{L_q}}\right\}, \quad \eta_y = \min\left\{\frac{1}{3\mu_y}, \frac{1}{3L_q\alpha}\right\}, \quad \eta_x = \frac{\mu_y}{\mu_x}\eta_y. \tag{21}$$

Additionally, the point $(\hat{x}^{k+1}, \hat{y}^{k+1})$ in line 5 should satisfy the following condition:

$$\left\| \begin{matrix} \nabla_x A_\eta^k(\hat{x}^{k+1}, \hat{y}^{k+1}) \\ -\nabla_y A_\eta^k(\hat{x}^{k+1}, \hat{y}^{k+1}) \end{matrix} \right\|^2_{\mathbf{P}^{-1}} \leq \frac{1}{6}\left\| \begin{matrix} \hat{x}^{k+1} - x^k \\ \hat{y}^{k+1} - y^k \end{matrix} \right\|^2_{\mathbf{P}}, \ \textit{where}\ \mathbf{P} := \begin{pmatrix} \frac{1}{\eta_x}\mathbf{I}_{d_x} & 0 \\ 0 & \frac{1}{\eta_y}\mathbf{I}_{d_y} \end{pmatrix}. \tag{22}$$

Then, for any

$$K \geq 3\max\left\{1, \sqrt{\tfrac{L_p}{\mu_x}}, \sqrt{\tfrac{L_q}{\mu_y}}\right\}\log\tfrac{C}{\varepsilon}$$

with C defined as follows

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

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

$$\frac{1}{\eta_x}\|x^K - x^*\|^2 + \frac{1}{\eta_y}\|y^K - y^*\|^2 \leq \varepsilon. \tag{23}$$

Proof of Theorem 1 you can find in Appendix A.1.

Complexity of solving auxiliary subproblems. In each iteration k of Algorithm 1, we are required to find a point $\left(\hat{x}^{k+1}, \hat{y}^{k+1}\right)$ that is an approximate solution to problem (19) and satisfies condition (22). According to Assumption 3, the function $\frac{1}{2\eta_x}\|x - x^k\|^2 + R(x, y) - \frac{1}{2\eta_y}\|y - y^k\|^2$ is smooth and strongly convex-strongly concave, while the composites $\langle\nabla p(x_g^k, x)\rangle$, $\langle\nabla q(y_g^k), y\rangle$ are proximal-friendly. The FOAM algorithm (Algorithm 4 from Kovalev & Gasnikov (2022)) is optimal for finding the solution to problem (19). This implies that it can be applied from the initial point (x^k, y^k) to find point $\left(\hat{x}^{k+1}, \hat{y}^{k+1}\right)$. The complexity of this procedure is outlined in the following theorem.

Theorem 2. *Consider the application of Algorithm 4 from Kovalev & Gasnikov (2022) to solve problem* (19) *under Assumption 3. Then, this algorithm requires the following number of gradient evaluations*

$$T = \mathcal{O}\left(\sqrt{(1 + L_R\eta_x)(1 + L_R\eta_y)}\log\tfrac{\max\{(1+L_R\eta_x)(1+L_R\eta_y)\}}{\min\{1/\eta_x, 1/\eta_y\}}\right) \tag{24}$$

to find a point $\left(\hat{x}^{k+1}, \hat{y}^{k+1}\right)$, that satisfies condition (22)*.*

Proof of Theorem 2 you can find in Appendix A.2.

Remark 1. *Note that the stopping criterion* (22) *for solving the auxiliary problem* (19) *is practical due to it does not depend on point $(\hat{x}_*^{k+1}, \hat{y}_*^{k+1})$ (solution to* (19)*).*

Overall complexity. We are now prepared to present the number of gradient computations for $\nabla p(x)$, $\nabla q(y)$, and $\nabla R(x, y)$, which are sufficient for finding a solution to problem (1) by Algorithm 1. It should be noted that finding the solution to the auxiliary subproblem (19) does not require the oracle calls of $\nabla p(x), \nabla q(y)$. These oracles are only called in line 6 of Algorithm 1. By combining the results from Theorems 1 and 2, we can present the following theorem.

Theorem 3. *Consider the application of Algorithm 1 to solve problem 1 under Assumptions 1-3, with tuning* (20) *or* (21)*. Also, to find point $(\hat{x}^{k+1}, \hat{y}^{k+1})$, that satisfy condition* (22)*, the FOAM algorithm will be used at each iteration of Algorithm 1. Then, this method requires*

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

gradient calls of $\nabla p(x)$, $\nabla q(y)$ and

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

gradient calls of $\nabla R(x, y)$ to find an ε-solution to problem (1)*.*

Proof of this theorem you can find in Appendix A.3.

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

3.3 Convex-concave and strongly convex-concave composite SPP

For the convex-concave composite saddle point problem, we assume that $\mu_x = \mu_y = 0$. This implies that the function $R(x, y)$ is convex-concave. Similarly, for the strongly convex-concave composite saddle point problem, we assume that $\mu_x > \mu_y = 0$, which means that the function $R(x, y)$ is strongly convex-concave. To present the results for these problems, we adopt the standard assumption that the solution (x^*, y^*) is bounded, i.e., $\|x^*\| \leq \mathcal{D}_x$, $\|y^*\| \leq \mathcal{D}_y$. We use this assumption and consider problem (1), which includes regularization terms. For the strongly convex-concave case, we apply regularization to the function $q(y)$ and consider the problem

$$\min_{x\in\mathbb{R}^{d_x}}\max_{y\in\mathbb{R}^{d_y}}\left\{p(x) + R(x, y) - q(y) - \frac{\varepsilon}{12\mathcal{D}_y^2}\|y\|^2\right\} \tag{25}$$

as opposed to the problem (1). Similarly, for the convex-concave case, we introduce regularization terms for the functions $p(x)$ and $q(y)$, and consider this modified problem

$$\min_{x\in\mathbb{R}^{d_x}}\max_{y\in\mathbb{R}^{d_y}}\left\{p(x) + \frac{\varepsilon}{16\mathcal{D}_x^2}\|x\|^2 + R(x, y) - q(y) - \frac{\varepsilon}{16\mathcal{D}_y^2}\|y\|^2\right\} \tag{26}$$

instead of problem (1). To demonstrate the equivalence of problems (25), (26) with regularization terms to problem (1), we present the following lemma.

Lemma 1. *Consider problem* (1) *under Assumptions 1 to 3. If $\mu_x > 0$, $\mu_y = 0$ (strongly convex-concave case) and $(\hat{x}, \hat{y})$ is a $\frac{2\varepsilon}{3}$-solution to problem* (25) *or if $\mu_x = 0$, $\mu_y = 0$ (convex-concave case) and $(\hat{x}, \hat{y})$ is a $\frac{\varepsilon}{2}$-solution to problem* (26) *with $\|x^*\| \leq \mathcal{D}_x$, $\|y^*\| \leq \mathcal{D}_y$. Then, $(\hat{x}, \hat{y})$ is an ε-solution to problem* (1).

Due to this lemma, we need to find a $\frac{2\varepsilon}{3}$-solution to problem (25) or a $\frac{\varepsilon}{2}$-solution to problem (26). To find them, we apply Algorithm 1 with composites $p(x)$, $q(y)$. According to Theorem 3, Algorithm 1 demands $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\sqrt{\mu_x\varepsilon}}\mathcal{D}_y\right\}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)$ gradient calls of $\nabla R(x, y)$ and $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\mu_x}}, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ gradient calls of $\nabla p(x), \nabla q(y)$ to find an ε-solution to problem (1) in the context of a strongly convex-concave case. Conversely, it requires $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y, \frac{L_R}{\varepsilon}\mathcal{D}_x\mathcal{D}_y\right\}\log\frac{L_R}{\min\{\mu_x,\mu_y\}}\log\frac{1}{\varepsilon}\right)$ gradient calls of $\nabla R(x, y)$ and $\mathcal{O}\left(\max\left\{\sqrt{\frac{L_p}{\varepsilon}}\mathcal{D}_x, \sqrt{\frac{L_q}{\varepsilon}}\mathcal{D}_y\right\}\log\frac{1}{\varepsilon}\right)$ gradient calls of $\nabla p(x), \nabla q(y)$ to find an ε-solution to (1) in a convex-concave scenario.

Remark 2. *Also, we analyze the important particular case of* (1)*, when $R(x, y) = x^T\mathbf{B}y$. Results for this case can be found in Appendix B.*

4 Discussion Our Results and Related Works

In this section, we compare complexity results for Algorithm 1 stated in Theorem 3 with related works.

4.1 Strongly convex-strongly concave and strongly convex-concave case

In the case of strong convexity and strong concavity, Algorithm 1 has the following oracle complexity

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

and

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

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

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

from Wang & Li (2020),

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

according to Lin et al. (2020),

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

from Kovalev & Gasnikov (2022) 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 Jin et al. (2022). It is important to note that the works Jin et al. (2022) have examined problem (1) under Assumptions 1, 2 and a more detailed assumption on the function $R(x,y)$.

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

Using this assumption, the authors of Jin et al. (2022) obtained the following iteration complexity for 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 achieves the lower bounds if $L_R^{xx}=L_R^{yy}=0$. Meanwhile, Algorithm 1 effectively separates the number of gradient calls of $\nabla p(x),\nabla q(y)$ from the number of gradient calls of $\nabla R(x,y)$ up to a logarithmic factor.

In work Alkousa et al. (2019), the authors separate the number of gradient calls of $\nabla p(x)$, $\nabla q(y)$ from the number of gradient calls of $\nabla R(x,y)$. However, these bounds do not achieve 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 effectively separates the number of gradient calls to (1). For the strongly convex-concave case, we get the same results with regularization by changing μ_y to $\varepsilon/\mathcal{D}_y^2$.

4.2 Convex-concave case

In the scenario of a convex-concave case, Algorithm 1 requires

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

and

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

to find an ε-solution to (1). This result meets 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 a logarithmic factor and generalizes results

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

and

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

from Lan & Ouyang (2021), where $\mathcal{D}=\max\{\mathcal{D}_x,\mathcal{D}_y\}$.

Remark 3. *The discussion of our results for the bilinear case can be found in Appendix C.*

REFERENCES

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.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Alfred Auslender and Marc Teboulle. Interior gradient and proximal methods for convex and conic optimization. siam journal on optimization. *SIAM Journal on Optimization*, 16(3):697–725, 2006.

Waïss Azizian, Damien Scieur, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. Accelerating smooth games by manipulating spectral shapes. 01 2020.

Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

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

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. *ArXiv*, abs/2005.14165, 2020.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40:120–145, 2011.

Yunmei Chen, Guanghui Lan, and Yuyuan Ouyang. Accelerated schemes for a class of variational inequalities. *Mathematical programming*, 165:113–149, 2017.

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

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.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International conference on machine learning*, pp. 1125–1134. PMLR, 2018.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, pp. 1049–1058. PMLR, 2017.

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.

F. Facchinei and J.S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2007. ISBN 9780387218151. URL `https://books.google.ru/books?id=lX_7Rce3_Q0C`.

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.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

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

Osman Guler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Optimization*, 29(2):403–419, 1991.

Adam Ibrahim, Waïss Azizian, Gauthier Gidel, and Ioannis Mitliagkas. Linear lower bounds and conditioning of differentiable games. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4583–4593. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/ibrahim20a.html.

Yujia Jin and Aaron Sidford. Efficiently solving MDPs with stochastic mirror descent. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pp. 4890–4900. PMLR, 2020.

Yujia Jin, Aaron Sidford, and Kevin Tian. Sharper rates for separable minimax and finite sum optimization via primal-dual extragradient methods. In Po-Ling Loh and Maxim Raginsky (eds.), *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 4362–4415. PMLR, 02–05 Jul 2022.

G. M. Korpelevich. The extragradient method for finding saddle points and other problems. 1976.

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.

Dmitry Kovalev, Adil Salim, and Peter Richtarik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18342–18352. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d530d454337fb09964237fecb4bea6ce-Paper.pdf.

Dmitry Kovalev, Alexander Gasnikov, and Peter Richtárik. Accelerated primal-dual gradient method for smooth and convex-concave saddle-point problems with bilinear coupling. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=FncDhRcRYiN.

Guanghui Lan. Gradient sliding for composite optimization. *Mathematical Programming*, 159(1): 201–235, 2016.

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

Adrian S Lewis and Stephen J Wright. A proximal method for composite minimization. *Mathematical programming*, 158(1-2):501–546, 2016.

Chris Junchi Li, Huizhuo Yuan, Gauthier Gidel, Quanquan Gu, and Michael Jordan. Nesterov meets optimism: rate-optimal separable minimax optimization. In *International Conference on Machine Learning*, pp. 20351–20383. PMLR, 2023.

Huan Li, Cong Fang, Wotao Yin, and Zhouchen Lin. Decentralized accelerated gradient methods with increasing penalty parameters. *IEEE Transactions on Signal Processing*, 68:4855–4870, 2020.

Tianyi Lin, Chi Jin, and Michael I. Jordan. Near-optimal algorithms for minimax optimization. *Thirty Third Conference on Learning Theory*, 125:2738–2779, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. *International Conference on Artificial Intelligence and Statistics*, pp. 1497–1507, 2019.

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. *SIAM Journal on Optimization*, 21(4):1688–1720, 2010.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. 1983.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Yurii Nesterov and Laura Scrimali. Solving strongly monotone variational and quasi-variational inequalities. 2006.

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. siam journal on control and optimization. *SIAM Journal on Optimization*, 14(5):877–898, 1976.

Alexander Rogozin, Demyan Yarmoshik, Ksenia Kopylova, and Alexander Gasnikov. Decentralized strongly-convex optimization with affine constraints: Primal and dual approaches. In *Advances in Optimization and Applications: 13th International Conference, OPTIMA 2022, Petrovac, Montenegro, September 26–30, 2022, Revised Selected Papers*, pp. 93–105. Springer, 2023.

Tim Roughgarden. Algorithmic game theory. *Communications of the ACM*, 53(7):78–86, 2010.

Adil Salim, Laurent Condat, Dmitry Kovalev, and Peter Richtarik. An optimal algorithm for strongly convex minimization under affine constraints. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4482–4498. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/salim22a.html.

Stefano Savazzi, Monica Nicoli, and Vittorio Rampa. Federated learning with cooperating devices: A consensus approach for massive iot networks. *IEEE Internet of Things Journal*, 7:4641–4654, 01 2020. doi: 10.1109/JIOT.2020.2964162.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

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

Ben Taskar, Simon Lacoste-Julien, and Michael Jordan. Structured prediction via the extragradient method. *Advances in neural information processing systems*, 18, 2005.

Kiran K. Thekumparampil, Niao He, and Sewoong Oh. Lifted primal-dual method for bilinearly coupled smooth minimax optimization. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4281–4308. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/thekumparampil22a.html.

P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *Journal on Control and Optimization*, 38 (2):431–446, 2000.

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

John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

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

Yuanhao Wang and Jian Li. Improved algorithms for convex-concave minimax optimization. *Neural Information Processing Systems 33 (NeurIPS 2020)*, 33:4800–4810, 2020.

Lin Xiao, Adams Wei Yu, Qihang Lin, and Weizhu Chen. Dscovr: Randomized primal-dual block coordinate algorithms for asynchronous distributed optimization. *Journal of Machine Learning Research*, 20(43):1–58, 2019. URL `http://jmlr.org/papers/v20/17-608.html`.

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

Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of machine learning research*, 10(7), 2009.

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

A MISSING PROOFS

A.1 PROOF OF THEOREM 1

Lemma 2. *Consider Algorithm 1 for Problem 1 under Assumptions 1-3 and* $\frac{L_p}{\mu_x} \geq \frac{L_q}{\mu_y}$*, with the following tuning:*

$$\alpha = \min\left\{1, \sqrt{\frac{\mu_x}{L_p}}\right\}, \quad \eta_x = \min\left\{\frac{1}{3\mu_x}, \frac{1}{3L_p\alpha}\right\}, \quad \eta_y = \frac{\mu_x}{\mu_y}\eta_x, \tag{27}$$

and let $(\hat{x}^{k+1}, \hat{y}^{k+1})$ *in line 5 satisfy*

$$\left\| \begin{matrix} \nabla_x A_\eta^k(\hat{x}^{k+1}, \hat{y}^{k+1}) \\ -\nabla_y A_\eta^k(\hat{x}^{k+1}, \hat{y}^{k+1}) \end{matrix} \right\|_{\mathbf{P}^{-1}}^2 \leq \frac{1}{6} \left\| \begin{matrix} \hat{x}^{k+1} - x^k \\ \hat{y}^{k+1} - y^k \end{matrix} \right\|_{\mathbf{P}}^2, \text{ where } \mathbf{P} := \begin{pmatrix} \frac{1}{\eta_x}\mathbf{I}_{d_x} & 0 \\ 0 & \frac{1}{\eta_y}\mathbf{I}_{d_y} \end{pmatrix}. \tag{28}$$

Then, the following inequality holds

$$\Psi^{k+1} \leq \left(1 - \frac{\alpha}{6}\right)\Psi^k, \tag{29}$$

where

$$\Psi^k := \frac{1}{\eta_x}\|x^k - x^*\|^2 + \frac{1}{\eta_y}\|y^k - y^*\|^2 + \frac{2}{\alpha}\mathrm{D}_p(x_f^k, x^*) + \frac{2}{\alpha}\mathrm{D}_q(y_f^k, y^*). \tag{30}$$

Proof. Using line 6 of Algorithm 1, we get

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

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

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

Using line 6 of Algorithm 1 and (19), we get

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

Since (28), the following inequality holds

$$\begin{aligned}
\left\| \begin{matrix} x^{k+1} - x^* \\ y^{k+1} - y^* \end{matrix} \right\|_{\mathbf{P}}^2 &\leq \left\| \begin{matrix} x^k - x^* \\ y^k - y^* \end{matrix} \right\|_{\mathbf{P}}^2 - \frac{2}{3}\left\| \begin{matrix} \hat{x}^{k+1} - x^k \\ \hat{y}^{k+1} - y^k \end{matrix} \right\|_{\mathbf{P}}^2 \\
&\quad - 2\left\langle \begin{matrix} \nabla p(x_g^k) + \nabla_x R(\hat{x}^{k+1}, \hat{y}^{k+1}); \hat{x}^{k+1} - x^* \\ \nabla q(y_g^k) - \nabla_y R(\hat{x}^{k+1}, \hat{y}^{k+1}); \hat{y}^{k+1} - y^* \end{matrix} \right\rangle.
\end{aligned}$$

Using the first-order necessary condition $\begin{pmatrix}\nabla p(x^*)\\ \nabla q(y^*)\end{pmatrix} + \begin{pmatrix}\nabla_x R(x^*,y^*)\\ -\nabla_y R(x^*,y^*)\end{pmatrix} = 0$ and Assumption 3, we get

$$\begin{aligned}
\left\|\begin{matrix}x^{k+1}-x^*\\ y^{k+1}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 &= \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{1}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2 - 2\left\langle\begin{matrix}\nabla p(x_g^k)-\nabla p(x^*)\\ \nabla q(y_g^k)-\nabla q(y^*)\end{matrix};\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\rangle\\
&\quad - 2\left\langle\begin{matrix}\nabla p(x_g^k)-\nabla p(x^*)\\ \nabla q(y_g^k)-\nabla q(y^*)\end{matrix};\begin{matrix}x^{k}-x_g^k\\ y^{k}-y_g^k\end{matrix}\right\rangle - 2\left\langle\begin{matrix}\nabla p(x_g^k)-\nabla p(x^*)\\ \nabla q(y_g^k)-\nabla q(y^*)\end{matrix};\begin{matrix}x_g^{k}-x^*\\ y_g^{k}-y^*\end{matrix}\right\rangle\\
&\quad - 2\left\langle\begin{matrix}\nabla_x R(\hat{x}^{k+1},\hat{y}^{k+1})-\nabla_x R(x^*,y^*)\\ -\nabla_y R(\hat{x}^{k+1},\hat{y}^{k+1})+\nabla_y R(x^*,y^*)\end{matrix};\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\rangle\\
&\le \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{2}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2 - 2\left\langle\begin{matrix}\nabla p(x_g^k)-\nabla p(x^*)\\ \nabla q(y_g^k)-\nabla q(y^*)\end{matrix};\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\rangle\\
&\quad - 2\left\langle\begin{matrix}\nabla p(x_g^k)-\nabla p(x^*)\\ \nabla q(y_g^k)-\nabla q(y^*)\end{matrix};\begin{matrix}x^{k}-x_g^k\\ y^{k}-y_g^k\end{matrix}\right\rangle - 2\left\langle\begin{matrix}\nabla p(x_g^k)-\nabla p(x^*)\\ \nabla q(y_g^k)-\nabla q(y^*)\end{matrix};\begin{matrix}x_g^{k}-x^*\\ y_g^{k}-y^*\end{matrix}\right\rangle\\
&\quad - \left\|\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\|_{\mathbf{M}}^2,
\end{aligned}$$

where $\mathbf{M} = \begin{pmatrix}\mu_x \mathbf{I}_{d_x} & 0\\ 0 & \mu_y \mathbf{I}_{d_y}\end{pmatrix}$. Now, we use line 4 and line 7 of Algorithm 1, and get

$$\begin{aligned}
\left\|\begin{matrix}x^{k+1}-x^*\\ y^{k+1}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 &\le \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{2}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{2}{\alpha}\left\langle\begin{matrix}\nabla p(x_g^k)\\ \nabla q(y_g^k)\end{matrix};\begin{matrix}x_f^{k+1}-x_g^k\\ y_f^{k+1}-y_g^k\end{matrix}\right\rangle\\
&\quad + \frac{2}{\alpha}\left\langle\begin{matrix}\nabla p(x^*)\\ \nabla q(y^*)\end{matrix};\begin{matrix}x_f^{k+1}-x^*\\ y_f^{k+1}-y^*\end{matrix}\right\rangle + \frac{2(1-\alpha)}{\alpha}\left\langle\begin{matrix}\nabla p(x_g^k)\\ \nabla q(y_g^k)\end{matrix};\begin{matrix}x_f^{k}-x_g^k\\ y_f^{k}-y_g^k\end{matrix}\right\rangle\\
&\quad - \frac{2(1-\alpha)}{\alpha}\left\langle\begin{matrix}\nabla p(x^*)\\ \nabla q(y^*)\end{matrix};\begin{matrix}x_f^{k}-x^*\\ y_f^{k}-y^*\end{matrix}\right\rangle - 2\left\langle\begin{matrix}\nabla p(x_g^k)\\ \nabla q(y_g^k)\end{matrix};\begin{matrix}x_g^{k}-x^*\\ y_g^{k}-y^*\end{matrix}\right\rangle - \left\|\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\|_{\mathbf{M}}^2.
\end{aligned}$$

Using Assumptions 1 and 2, we get

$$\begin{aligned}
\left\|\begin{matrix}x^{k+1}-x^*\\ y^{k+1}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 &\le \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{2}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2 + \frac{2}{\alpha}\left(p(x_g^k)-p(x_f^{k+1}) + \frac{L_p}{2}\|x_f^{k+1}-x_g^k\|^2\right.\\
&\quad \left. + q(y_g^k)-q(y_f^{k+1}) + \frac{L_q}{2}\|y_f^{k+1}-y_g^k\|^2 + \left\langle\begin{matrix}\nabla p(x^*)\\ \nabla q(y^*)\end{matrix};\begin{matrix}x_f^{k+1}-x^*\\ y_f^{k+1}-y^*\end{matrix}\right\rangle\right)\\
&\quad + \frac{2(1-\alpha)}{\alpha}\left(p(x_f^k)-p(x_g^k)+q(y_f^k)-q(y_g^k) - \left\langle\begin{matrix}\nabla p(x^*)\\ \nabla q(y^*)\end{matrix};\begin{matrix}x_f^{k}-x^*\\ y_f^{k}-y^*\end{matrix}\right\rangle\right)\\
&\quad + 2\left(p(x^*)-p(x_g^k)+q(y^*)-q(y_g^k)\right) - \left\|\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\|_{\mathbf{M}}^2\\
&= \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{2}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{2}{\alpha}\left(\mathrm{D}_p(x_f^{k+1},x^*)+\mathrm{D}_q(y_f^{k+1},y^*)\right)\\
&\quad + \left\|\begin{matrix}x_f^{k+1}-x_g^k\\ y_f^{k+1}-y_g^k\end{matrix}\right\|_{\frac{1}{\alpha}\mathbf{L}}^2 + \frac{2(1-\alpha)}{\alpha}\left(\mathrm{D}_p(x_f^k,x^*)+\mathrm{D}_q(y_f^k,y^*)\right) - \left\|\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\|_{\mathbf{M}}^2,
\end{aligned}$$

where $\mathbf{L} = \begin{pmatrix}L_p \mathbf{I}_{d_x} & 0\\ 0 & L_q \mathbf{I}_{d_y}\end{pmatrix}$. By (27) and condition $\frac{L_p}{\mu_x} \ge \frac{L_q}{\mu_y}$ the following inequalities hold

$$\eta_x \le \frac{1}{3L_p\alpha}, \quad \eta_y = \frac{\mu_x}{\mu_y}\eta_x \le \frac{\mu_x}{\mu_y}\cdot\frac{1}{3L_p\alpha} \le \frac{1}{3L_q\alpha}.$$

Using these inequalities and line 7 of Algorithm 1, we get

$$\left\|\begin{matrix}x_f^{k+1}-x_g^k\\ y_f^{k+1}-y_g^k\end{matrix}\right\|_{\frac{1}{\alpha}\mathbf{L}}^2 = \frac{1}{\alpha}\left\|\begin{matrix}x_f^{k+1}-x_g^k\\ y_f^{k+1}-y_g^k\end{matrix}\right\|_{\mathbf{L}}^2 = \alpha\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{L}}^2 \le \frac{1}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2.$$

Apply this inequality to estimate $\left\|\begin{matrix}x^{k+1}-x^*\\ y^{k+1}-y^*\end{matrix}\right\|_{\mathbf{P}}^2$, we get

$$\left\|\begin{matrix}x^{k+1}-x^*\\ y^{k+1}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 \leq \left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{1}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2 - \frac{2}{\alpha}\left(\mathrm{D}_p(x_f^{k+1},x^*)+\mathrm{D}_q(y_f^{k+1},y^*)\right) + \frac{2(1-\alpha)}{\alpha}\left(\mathrm{D}_p(x_f^{k},x^*)+\mathrm{D}_q(y_f^{k},y^*)\right) - \left\|\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\|_{\mathbf{M}}^2.$$

Since (27), the following inequalities hold $\eta_x \leq \frac{1}{3\mu_x}$, $\eta_y = \frac{\mu_x}{\mu_y}\eta_x \leq \frac{1}{3\mu_y}$. Using these inequalities and $\|a-b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, we get

$$\begin{aligned}\frac{1}{3}\left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{P}}^2 + \left\|\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\|_{\mathbf{M}}^2 &\geq \left\|\begin{matrix}\hat{x}^{k+1}-x^k\\ \hat{y}^{k+1}-y^k\end{matrix}\right\|_{\mathbf{M}}^2 + \left\|\begin{matrix}\hat{x}^{k+1}-x^*\\ \hat{y}^{k+1}-y^*\end{matrix}\right\|_{\mathbf{M}}^2\\ &\geq \frac{1}{2}\left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{M}}^2 \geq \frac{\alpha}{6}\left\|\begin{matrix}x^{k}-x^*\\ y^{k}-y^*\end{matrix}\right\|_{\mathbf{P}}^2.\end{aligned}$$

In the last inequality we use the following inequality $\eta_y\mu_y = \eta_x\mu_x \geq \frac{\alpha}{3}$, that follows by (27):

1. If $L_p \leq \mu_x,$ then $\alpha = 1, \eta_x\mu_x = \frac{1}{3} = \frac{\alpha}{3}, \eta_y\mu_y = \frac{1}{3} = \frac{\alpha}{3}$.
2. If $L_p > \mu_x,$ then $\alpha = \sqrt{\frac{\mu_x}{L_p}}, \eta_x\mu_x = \sqrt{\frac{\mu_x}{3L_p}} \geq \frac{\alpha}{3}, \eta_y\mu_y = \sqrt{\frac{\mu_x}{3L_p}} \geq \frac{\alpha}{3}$.

By definition of Ψ^k (30), we get

$$\Psi^{k+1} \leq \left(1-\frac{\alpha}{6}\right)\Psi^k.$$

□

Proof of Theorem 1 Using the property of the Bregman divergence for a differentiable convex function $\mathrm{D}_f(x,y) \geq 0$ and running the recursion (29) we get

$$\frac{1}{\eta_x}\|x^K - x^*\|^2 + \frac{1}{\eta_y}\|y^K - y^*\|^2 \leq \Psi^K \leq C\left(1-\frac{\alpha}{6}\right)^K,$$

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 $K = \frac{6}{\alpha}\log\frac{1}{\varepsilon}$ iterations of Algorithm 1 we get a pair (x^K, y^K) satisfies the following inequality

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

A.2 Proof of Theorem 2

Lemma 3. *Consider problem* (19) *under Assumption 3 and a point* $\left(\hat{x}^{k+1}, \hat{y}^{k+1}\right)$*, that is a* γ*-solution to this problem with* γ *defined as follows*

$$\gamma = C\left\|\begin{matrix}x^{k}-\hat{x}_*^{k+1}\\ y^{k}-\hat{y}_*^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2, \tag{31}$$

where

$$C = \left(12\max\left\{\left(L_R + \frac{1}{\eta_x}\right)^2\eta_x, \left(L_R + \frac{1}{\eta_y}\right)^2\eta_y\right\} + 2\max\left\{\frac{1}{\eta_x}, \frac{1}{\eta_y}\right\}\right)^{-1}. \tag{32}$$

Then, point $\left(\hat{x}^{k+1}, \hat{y}^{k+1}\right)$ *satisfies condition* (22).

Proof. Let $\left(\hat{x}^{k+1}, \hat{y}^{k+1}\right)$ is γ-solution to problem (19). Then, using this fact and inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, we get

$$\begin{aligned}\left\|\begin{matrix}\hat{x}^{k+1} - \hat{x}_*^{k+1} \\ \hat{y}^{k+1} - \hat{y}_*^{k+1}\end{matrix}\right\|^2 &\leq C\left\|\begin{matrix}x^k - \hat{x}_*^{k+1} \\ y^k - \hat{y}_*^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2 \leq 2C\left\|\begin{matrix}x^k - \hat{x}^{k+1} \\ y^k - \hat{y}^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2 + 2C\left\|\begin{matrix}\hat{x}^{k+1} - \hat{x}_*^{k+1} \\ \hat{y}^{k+1} - \hat{y}_*^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2 \\ &\leq 2C\left\|\begin{matrix}x^k - \hat{x}^{k+1} \\ y^k - \hat{y}^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2 + 2C\lambda_{\max}(\mathbf{P})\left\|\begin{matrix}\hat{x}^{k+1} - \hat{x}_*^{k+1} \\ \hat{y}^{k+1} - \hat{y}_*^{k+1}\end{matrix}\right\|^2,\end{aligned}$$

where the point $\left(\hat{x}_*^{k+1}, \hat{y}_*^{k+1}\right)$ is solution to problem (19) and $\lambda_{\max}(\mathbf{A})$ is the maximum eigenvalue of matrix $\mathbf{A}$. After rearranging, we get

$$\left\|\begin{matrix}\hat{x}^{k+1} - \hat{x}_*^{k+1} \\ \hat{y}^{k+1} - \hat{y}_*^{k+1}\end{matrix}\right\|^2 \leq \frac{2C}{1 - 2C\lambda_{\max}(\mathbf{P})}\left\|\begin{matrix}x^k - \hat{x}^{k+1} \\ y^k - \hat{y}^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2.$$

Using this fact, $\left(L_R + \frac{1}{\eta_x}\right)$ smoothness of $A_\eta^k(\cdot, y)$ and $\left(L_R + \frac{1}{\eta_y}\right)$ smoothness of $A_\eta^k(x, \cdot)$, and the first optimal condition for problem (19), and (32), we obtain the following inequality

$$\begin{aligned}\left\|\begin{matrix}\nabla_x A_\eta^k(\hat{x}^{k+1}, \hat{y}^{k+1}) \\ -\nabla_y A_\eta^k(\hat{x}^{k+1}, \hat{y}^{k+1})\end{matrix}\right\|_{\mathbf{P}^{-1}}^2 &\leq \left\|\begin{matrix}\hat{x}^{k+1} - \hat{x}_*^{k+1} \\ \hat{y}^{k+1} - \hat{y}_*^{k+1}\end{matrix}\right\|_{\mathbf{L}_\eta^2\mathbf{P}^{-1}}^2 \leq \lambda_{\max}(\mathbf{L}_\eta^2\mathbf{P}^{-1})\left\|\begin{matrix}\hat{x}^{k+1} - \hat{x}_*^{k+1} \\ \hat{y}^{k+1} - \hat{y}_*^{k+1}\end{matrix}\right\|^2 \\ &\leq \frac{2C\lambda_{\max}(\mathbf{L}_\eta^2\mathbf{P}^{-1})}{1 - 2C\lambda_{\max}(\mathbf{P})}\left\|\begin{matrix}x^k - \hat{x}^{k+1} \\ y^k - \hat{y}^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2 = \frac{1}{6}\left\|\begin{matrix}x^k - \hat{x}^{k+1} \\ y^k - \hat{y}^{k+1}\end{matrix}\right\|_{\mathbf{P}}^2,\end{aligned}$$

where $\mathbf{L}_\eta = L_R\mathbf{I}_{d_x+d_y} + \mathbf{P}$. □

Lemma 4. *Consider the function $R(x, y)$ under Assumption 3 and replace variables as $x = \alpha u$, $y = \beta v$. Then function $\tilde{R}(u, v) := R(x, y)$ is $\tilde{L}$-smooth, function $\tilde{R}(\cdot, v)$ is μ_u-strongly convex and function $\tilde{R}(u, \cdot)$ is μ_v-strongly concave, with $\tilde{L} = \max\{\alpha^2, \beta^2\}L$, $\mu_u = \alpha^2\mu_x$, $\mu_v = \beta^2\mu_y$.*

Proof. Firstly, let us consider that

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

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

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

with $\tilde{L} = \max\{\alpha^2, \beta^2\}L$. Now we define μ_u-strong convexity of function $\tilde{R}(\cdot, v)$.

$$\begin{aligned}\tilde{R}(u_2, v) = R(x_2, y) &\geq R(x_1, y) + \langle\nabla_x R(x_1, y), x_2 - x_1\rangle + \frac{\mu_x}{2}\|x_2 - x_1\|^2 \\ &= \tilde{R}(u_1, v) + \left\langle\frac{1}{\alpha}\nabla_u\tilde{R}(u_1, v), \alpha(u_2 - u_1)\right\rangle + \frac{\mu_x\alpha^2}{2}\|u_2 - u_1\|^2 \\ &= \tilde{R}(u_1, v) + \left\langle\nabla_u\tilde{R}(u_1, v), u_2 - u_1\right\rangle + \frac{\mu_u}{2}\|u_2 - u_1\|^2,\end{aligned}$$

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

Proof of Theorem 2. By Lemma 3, to find a point $\left(\hat{x}^{k+1}, \hat{y}^{k+1]}\right)$, that satisfies condition (22), we need to find a γ-solution to problem (19), where γ defined in (31). Firstly, we make the following replacing variables $x = \alpha u$, $y = \beta v$ in the problem (19). After that we get the following problem in new variables:

$$\min_u \max_v \alpha\langle \nabla p(x_g^k), u\rangle + \frac{\alpha^2}{2\eta_x}\left\|u - \frac{x^k}{\alpha}\right\|^2 + \tilde{R}(u,v) - \beta\langle \nabla q(y_g^k), v\rangle - \frac{\beta^2}{2\eta_y}\left\|v - \frac{y^k}{\beta}\right\|^2. \quad (33)$$

By Corollary 1 from Kovalev & Gasnikov (2022) the FOAM algorithm requires the following number of gradient evaluations:

$$T = \mathcal{O}\left(\frac{\tilde{L}_R + \frac{\alpha^2}{\eta_x} + \frac{\beta^2}{\eta_y}}{\sqrt{(\mu_u + \frac{\alpha^2}{\eta_x})(\mu_v + \frac{\beta^2}{\eta_y})}}\log\frac{1}{\gamma}\right) \quad (34)$$

to find a γ-solution to problem (33). By Lemma 4 we get $\tilde{L}_R = \max\{\alpha^2, \beta^2\}L_R$, $\mu_u = \alpha^2\mu_x$ and $\mu_v = \beta^2\mu_y$. Using these values we get the following number of gradient evaluations:

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

Now we are ready to define constants α, β: $\alpha = \sqrt{L_R + \frac{1}{\eta_y}}$, $\beta = \sqrt{L_R + \frac{1}{\eta_x}}$ and get

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

The choice of γ in (31) completes the proof.

A.3 Proof of Theorem 3

It should be noted that finding the solution to the auxiliary subproblem (19) does not require the gradient calls of $\nabla p(x), \nabla q(y)$. These gradients are only called in line 6 of Algorithm 1. This implies that the number of gradient calls for $\nabla p(x), \nabla q(y)$ is equivalent to K, as defined in Theorem 1. However, the gradient calls of $\nabla R(x, y)$ are essential for finding the solution to problem (19). This implies that the total number of gradient calls for $\nabla R(x, y)$ is $K \times T$, with T defined in Theorem 3. To estimate this, we perform some mathematical calculations for the case where $\frac{L_p}{\mu_x} \geq \frac{L_q}{\mu_y}$.

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

The case $\frac{L_q}{\mu_y} > \frac{L_p}{\mu_x}$ is symmetric.

B BILINEAR SADDLE POINT PROBLEMS

In the special case, when $R(x, y) = x^T \mathbf{B} y$, (1) has been also widely studied, dating at least to the classic work of Chambolle & Pock (2011) (imaging inverse problems). Modern applications can be found in decentralized optimization Rogozin et al. (2023); Chezhegov et al. (2023). Quadratic variant of the problem (1) also appeared in reinforcement learning Du et al. (2017). In this section we present our results for bilinear saddle point problems.

B.1 STRONGLY CONVEX-STRONGLY CONCAVE BILINEAR SPP

The bilinear strongly convex-strongly concave problem has the following form

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} p(x) + x^T \mathbf{B} y - q(y). \tag{35}$$

To this problem we assume that the following assumptions hold

Assumption 5. *$p(x) : \mathbb{R}^{d_x} \to \mathbb{R}$ is L_p-smooth and μ_p-strongly convex function*

Assumption 6. *$q(y) : \mathbb{R}^{d_y} \to \mathbb{R}$ is L_q-smooth and μ_q-strongly convex function*

Assumption 7. *Matrix $\mathbf{B} \in \mathbb{R}^{d_x \times d_y}$ is positive semi-definite.*

To apply Algorithm 1 to the problem (35) we reformulate it as a problem

$$\min_x \max_y \tilde{p}(x) + \frac{\mu_p}{2}\|x\|^2 + x^T B y - \frac{\mu_q}{2}\|y\|^2 - \tilde{q}(y)$$

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

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

$$\min_x \max_y \langle \nabla \tilde{p}(x_g^k), x\rangle + \frac{1}{2\eta_x}\|x - x^k\|^2 + \frac{\mu_p}{2}\|x\|^2 + x^T B y - \frac{\mu_q}{2}\|y\|^2 - \frac{1}{2\eta_y}\|y - y^k\|^2 - \langle \nabla \tilde{q}(y_g^k), y\rangle \tag{36}$$

with γ defined in (31). The simplest way to solve this problem is to reformulate it as a minimization problem in x using the first order optimal condition in y:

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

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

After reformulation, we get the quadratic problem

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

with

$$A = \frac{1}{2}\left(\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right) I + BB^T\right),$$

$$b = \nabla \tilde{p}(x_g^k) - \frac{1}{\eta_y} x^k + \left(1 - \frac{2\eta_y}{1 + \eta_y \mu_q}\right) B\left(\nabla \tilde{q}(y_g^k) - \frac{1}{\eta_y} y^k\right),$$

$$c = \frac{\eta_y}{2(1 + \eta_y \mu_q)}\|\nabla \tilde{q}(y_g^k)\|^2 - \frac{1}{1 + \mu_q \eta_y}\langle \nabla \tilde{q}(y_g^k), y^k\rangle + \frac{\mu_q(1 - \eta_y \mu_q)}{2(1 + \eta_y \mu_q)^2}\|y^k\|^2.$$

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

$$T = \mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}}\log\frac{1}{\gamma}\right) = \mathcal{O}\left(\sqrt{\frac{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right) + \lambda_{\max}(\mathbf{B}\mathbf{B}^T)}{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right) + \lambda_{\min}(\mathbf{B}\mathbf{B}^T)}}\log\frac{1}{\gamma}\right)$$

$$= \mathcal{O}\left(\min\left\{\sqrt{\frac{\lambda_{\max}(\mathbf{B}\mathbf{B}^T)}{\lambda_{\min}(\mathbf{B}\mathbf{B}^T)}}, \sqrt{1 + \frac{\lambda_{\max}(\mathbf{B}\mathbf{B}^T)}{\left(\frac{1}{\eta_x} + \mu_p\right)\left(\frac{1}{\eta_y} + \mu_q\right)}}\right\}\log\frac{\sqrt{\lambda_{\max}(\mathbf{B}\mathbf{B}^T)}}{\min\{\mu_p, \mu_q\}}\right)$$

iterations or calls of oracles B/B^T to find an ε-solution to (36).

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

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

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

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

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

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 (35).

Theorem 4. *Consider Problem* (35) *under Assumptions 5 to 7. 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

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

where

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

and

$$K_2 = \max\left\{\sqrt{\frac{L_p}{\mu_p}}, \sqrt{\frac{L_q}{\mu_q}}, \sqrt{\frac{\lambda_{\max}(\mathbf{B}\mathbf{B}^T)}{\mu_p\mu_q}}\right\}.$$

B.2 Affinely constrained minimization

This problem has the following form:

$$\min_{\mathbf{B}x=c} p(x), \tag{37}$$

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

$$\min_x \max_y p(x) + x^T\mathbf{B}y - y^Tc. \tag{38}$$

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

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

By Lemma 1, if we find $\frac{2\varepsilon}{3}$-solution to this problem, then we find an ε-solution to (38). To find this solution we apply Algorithm 1 and get the following complexity.

Corollary 1. *Consider Problem* (38)*. Then, to find an ε-solution, Algorithm 1 requires*

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

and

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

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

B.3 Bilinear problem with linear composites

In this subsection we consider bilinear problem with linear composites:

$$\min_x \max_y x^T d + x^T \mathbf{B} y - y^T c, \tag{39}$$

where matrix $\mathbf{B}$ is positive definite ($\lambda_{\min}(\mathbf{B}\mathbf{B}^T) = \lambda^+_{\min}(\mathbf{B}\mathbf{B}^T)$). As in the previous subsection, we make the regularization to apply Algorithm 1. The problem (39) with regularization has the following form:

$$\min_x \max_y \frac{\varepsilon}{16\mathcal{D}_x^2}\|x\|^2 + x^T d + x^T \mathbf{B} y - y^T c - \frac{\varepsilon}{16\mathcal{D}_y^2}\|y\|^2. \tag{40}$$

We need to find a $\frac{\varepsilon}{2}$-solution to find an ε-solution to (39) by Lemma 1. To find it we apply Algorithm 1 with the following complexity.

Corollary 2. *Consider Problem* (39)*. Then, to find an ε-solution, Algorithm 1 requires*

$$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{B}\mathbf{B}^T)}{\lambda_{\min}(\mathbf{B}\mathbf{B}^T)}}\log^2\frac{1}{\varepsilon}\right) \text{ calls of } B \text{ or } B^T.$$

C Discussion Our Results for Bilinear Problem and Related Works

C.1 Bilinear strongly convex-strongly concave case

For the bilinear strongly convex-strongly concave case (35) Algorithm 1 requires

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

and

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

to find an ε-solution to (35). 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 Kovalev et al. (2022), Thekumparampil et al. (2022), Du et al. (2022) but the main benefit of our approach is complexity separation.

C.2 AFFINELY CONSTRAINED MINIMIZATION CASE

For the affinely constrained minimization case (38) Algorithm 1 requires

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

and

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

This matches the iteration complexity of algorithms from the works Kovalev et al. (2020), Kovalev et al. (2022) up to logarithmic factor. Note, in these works, the authors achieve the lower bounds Salim et al. (2022) exactly. But the key idea of Algorithm 1 in separating oracle complexities.

Meanwhile, we can apply this results to distributed optimization problem (7). 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 Scaman et al. (2017) up to logarithmic factor. The optimal method for this problem was proposed in Beznosikov et al. (2020).

C.3 BILINEAR CASE WITH LINEAR COMPOSITES

For the bilinear case with linear composites (39) Algorithm 1 requires

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

and

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

These results match the iteration complexity from the work Azizian et al. (2020) up to logarithmic factor. In contrast to our results, in work Azizian et al. (2020) the lower bounds Ibrahim et al. (2020) are achieved.