Tight Complexity Bounds and Optimality Conditions for First-Order and Proximal Algorithms in Convex-Concave Saddle Point Problems

Chris Junchi Li[⋄]

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

October 7, 2024

Abstract

This paper explores the lower iteration complexity bounds for solving convex-concave minimax optimization problems using first-order methods. We analyze two classes of algorithms: pure first-order methods and proximal algorithms. By constructing worst-case examples, we establish tight lower bounds on the iteration complexity for these algorithm classes under different parameter regimes. These bounds provide theoretical insights into the computational limits of solving saddle point problems with strongly convex-concave structures. The results extend known lower bounds and have implications for real-world applications such as game theory, machine learning, and image processing.

Keywords: Convex-Concave Optimization, Min-Max Problem, First-Order Methods, Proximal Algorithms, Iteration Complexity.

1 Introduction

Minimax optimization problems are of fundamental importance in various fields such as game theory, adversarial machine learning, and imaging. The goal is to find a saddle point for the following problem:

$$\min_x \max_y F(x, y) \tag{1}$$

This problem arises in numerous applications, such as game theory [von Neumann et al., 2007, Nisan et al., 2007], image deconvolution [Chambolle & Pock, 2011], parallel computing [Xiao et al., 2019], adversarial training [Goodfellow et al., 2014, Arjovsky et al., 2017], and statistical learning [Abadeh et al., 2015]. Many practical algorithms, including gradient-based methods, belong to the class of first-order methods due to their scalability in large-dimensional settings.

In this paper, we establish tight lower iteration complexity bounds for first-order methods in convex-concave min-max saddle point problems. We analyze both pure first-order and proximal algorithms, deriving tight lower bounds through worst-case examples under different parameter regimes. Additionally, we provide theoretical insights into the computational limits of these algorithms and discuss their broader impact on real-world applications, such as adversarial learning.

To proceed, let us introduce the following two problem classes.

Definition 1 (Problem class $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$)**.** *$F(\cdot, y)$ is μ_x-strongly convex for any fixed y and $F(x, \cdot)$ is μ_y-strongly concave for any fixed x. Overall, the function F is smooth and ∇F*

satisfies the following Lipschitz continuity condition

$$\begin{cases} \|\nabla_x F(x_1, y) - \nabla_x F(x_2, y)\| \le L_x \|x_1 - x_2\|, & \forall x_1, x_2, y \\ \|\nabla_y F(x, y_1) - \nabla_y F(x, y_2)\| \le L_y \|y_1 - y_2\|, & \forall x, y_1, y_2 \\ \|\nabla_x F(x, y_1) - \nabla_x F(x, y_2)\| \le L_{xy} \|y_1 - y_2\|, & \forall x, y_1, y_2 \\ \|\nabla_y F(x_1, y) - \nabla_y F(x_2, y)\| \le L_{xy} \|x_1 - x_2\|, & \forall x_1, x_2, y. \end{cases} \tag{2}$$

We shall remark here that the constants in (2) may also be understood as the bounds on the different blocks of the Hessian matrix $\nabla^2 F(x, y)$ if F is twice continuously differentiable. That is,

$$\sup_{x,y} \|\nabla^2_{xx} F(x, y)\|_2 \le L_x, \quad \sup_{x,y} \|\nabla^2_{yy} F(x, y)\|_2 \le L_y, \quad \sup_{x,y} \|\nabla^2_{xy} F(x, y)\|_2 \le L_{xy}$$

However, throughout this paper we do not assume either $F(\cdot, y)$ or $F(x, \cdot)$ is second-order differentiable.

The second problem class is the bilinear saddle point model:

Definition 2 (Bilinear class $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$)**.** *In this special class, the problems are written as*

$$\min_x \max_y F(x, y) := f(x) + x^\top A y - g(y) \tag{3}$$

where $f(x)$ and $g(y)$ are both lower semi-continuous with $f(x)$ being μ_x-strongly convex and $g(y)$ being μ_y-strongly convex. The coupling matrix A satisfies $\|A\|_2 \le L_{xy}$.

For this special model class $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$, we assume the availability of the following prox-operations:

$$\text{prox}_{\gamma f}(v) := \operatorname*{argmin}_x f(x) + \frac{1}{2\gamma}\|x - v\|^2 \quad \text{and} \quad \text{prox}_{\sigma g}(u) := \operatorname*{argmin}_y g(y) + \frac{1}{2\sigma}\|y - u\|^2 \tag{4}$$

In this paper we shall establish the lower iteration complexity bound

$$\Omega\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L_{xy}^2}{\mu_x \mu_y} + \frac{L_y}{\mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right) \text{ for } \mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$$

and

$$\Omega\left(\sqrt{\frac{L_{xy}^2}{\mu_x \mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right) \text{ for } \mathcal{B}(L_{xy}, \mu_x, \mu_y)$$

with the proximal oracles (4). In particular, we first establish these lower bounds for pure first-order and general proximal algorithm classes under the linear span assumption. Later on we generalize the results for more general algorithm classes without the linear span assumption through the orthogonal invariance technique introduced by [Nemirovsky, 1992]. For more detailed applications of the orthogonal invariance technique in the lower bound derivation, the interested readers are referred to [Carmon et al., 2017, Carmon et al., 2019, Ouyang & Xu, 2018].

As an application of the above bound, we apply proper scaling to the worst-case instances and show that the above result implies several exisiting lower bounds for general convex-concave problems with bounded saddle point solutions. Specifically, we have

$$\Omega\left(\sqrt{\frac{L_x R_x^2}{\epsilon}} + \frac{L_{xy} R_x R_y}{\epsilon} + \sqrt{\frac{L_y R_y^2}{\epsilon}}\right) \text{ for } \mathcal{F}(L_x, L_y, L_{xy}, 0, 0), \text{ and } \|x^*\| \le R_x, \|y^*\| \le R_y$$

and

$$\Omega\left(\frac{L_{xy}R_xR_y}{\epsilon}\right) \text{ for } \mathcal{B}(L_{xy},0,0), \text{ and } \|x^*\| \le R_x, \|y^*\| \le R_y$$

For the above two lower bounds, we remark that under specific parameter regimes, the first bound is known, see [Nemirovsky, 1992] for the case with $L_x = L_y = L_{xy}$ and see [Ouyang & Xu, 2018] for the case with $L_y = 0$. However to our best knowledge, the first bound under a general set of parameters as well as the second bound for bilinear problem class are not known. Similar reductions can also be done for the problem classes with only one of μ_x and μ_y equal to 0, for which the lower bounds have already been discovered in [Ouyang & Xu, 2018].

Such lower iteration complexity results shed light on understanding the performance of the algorithms designed for min-max saddle point models. There are numerous results in the literature prior to ours. As a special case of (1), the lower bound results of convex minimization problem with $F(x,y) = f(x)$ has been well-studied in the past decades. For convex problems, Nesterov's accelerated gradient method have achieved iteration complexities of $\mathcal{O}(\sqrt{L/\epsilon})$ for L-smooth convex problems, and $\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\cdot\log\left(\frac{1}{\epsilon}\right)\right)$ for L-smooth and μ-strongly convex problems respectively, and both of them are shown to match the lower complexity bound for the first-order methods; see [Nesterov, 2018b].

However, for the min-max saddle-point models, the situation is more subtle. Due to the convex-concave nature of F, the vector field

$$G(x,y) = \begin{pmatrix} \nabla_x F(x,y) \\ -\nabla_y F(x,y) \end{pmatrix}$$

is monotone. Hence the convex-concave saddle point problem is often studied as a subclass of the variational inequality problems (VIP); see e.g. [Nesterov, 2007, Nesterov & Scrimali, 2006, Juditsky et al., 2011, Nemirovski, 2004, Marcotte & Dussault, 1987, Taji et al., 1993] and references therein. Although there have been plenty of studies on the variational inequalities model, the roles played by different Lischitz constants on the different blocks of variables have not been fully explored in the literature. In other words, often one would denote L to be an overall Lipschitz constant of the vector field G, which is of the order $\Theta(\max\{L_x, L_y, L_{xy}\})$ in our case, and set μ to be the strong monotonicity parameter of G, which is of the order $\Theta(\min\{\mu_x, \mu_y\})$ in our case, and no further distinctions among the parameters would be made. Hence the considered problems are of special instances in $\mathcal{F}(L, L, L, \mu, \mu)$. Under such settings, many algorithms including the mirror-prox algorithm [Nemirovski, 2004], the extra-gradient methods [Korpelevich, 1976, Mokhtari et al., 2019], and the accelerated dual extrapolation[1] [Nesterov & Scrimali, 2006] and so on, have all achieved the iteration complexity of $\mathcal{O}\left(\frac{L}{\mu}\cdot\log\left(\frac{1}{\epsilon}\right)\right)$, and this complexity is shown to be optimal for first-order methods in solving the problem class $\mathcal{F}(L, L, L, \mu, \mu)$; see [Nemirovsky & Yudin, 1983]. However, under the more general parameter regime of $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$, these methods are not optimal. For example, Nesterov's accelerated dual extrapolation method [Nesterov & Scrimali, 2006] has a complexity of $\mathcal{O}\left(\frac{\max\{L_x, L_{xy}, L_y\}}{\min\{\mu_x, \mu_y\}}\cdot\log\left(\frac{1}{\epsilon}\right)\right)$, even if the algorithm are modified carefully one can only guarantee a complexity of $\mathcal{O}\left(\sqrt{\frac{L_x^2}{\mu_x^2}+\frac{L_{xy}^2}{\mu_x\mu_y}+\frac{L_y^2}{\mu_y^2}}\cdot\log\left(\frac{1}{\epsilon}\right)\right)$, both of which do not match the lower bound provided in this paper. More recently, tighter upper bounds have been derived. In [Lin et al., 2020], the authors consider the problems class of $\mathcal{F}(L, L, L, \mu_x, \mu_y)$ and achieve an upper bound of $\mathcal{O}\left(\sqrt{\frac{L^2}{\mu_x\mu_y}}\cdot\log^3\left(\frac{1}{\epsilon}\right)\right)$, which matches our lower bound when $L_x = L_y = L_{xy} = L$ up

[1] In Nesterov's original paper [Nesterov & Scrimali, 2006], the author did not give a name to his algorithm. For convenience of referencing, in this paper we shall call it *accelerated dual extrapolation*.

to a logarithmic term. In [Wang & Li, 2020], the authors consider the general problems class of $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$, the proposed algorithm achieves an upper bound of $\mathcal{O}\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L \cdot L_{xy}}{\mu_x \mu_y} + \frac{L_y}{\mu_y}} \cdot \log^3\left(\frac{L^2}{\mu_x \mu_y}\right) \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ where $L = \max\{L_x, L_y, L_{xy}\}$, which almost matches our lower bound for the general problem class. Despite the gap for the general problem class $\mathcal{F}(L_x, L_{xy}, L_y, \mu_x, \mu_y)$, given the availability of proximal operators the authors of [Chambolle & Pock, 2011, Chambolle & Pock, 2016] have derived an algorithm for problem class $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$ with complexity $\mathcal{O}\left(\sqrt{\frac{L_{xy}^2}{\mu_x \mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$. We will prove in this paper that this result has matched the theoretical lower complexity bound for its problem and algorithm classes, hence optimal.

For the bilinear problem (3), when f is smooth and convex, $g(y) = b^\top y$ is linear, the problem is equivalent to the following convex optimization problem

$$\min_x \{f(x) : A^\top x - b = 0\}$$

Without using projection onto the hyperplane $\{x : A^\top x = b\}$ which requires a matrix inversion, pure first-order methods achieve $\mathcal{O}(1/\epsilon)$ complexity despite the strong convexity of f; see e.g. [Gao & Zhang, 2017, Xu, 2017, Ouyang et al., 2015]. Those iteration complexity bounds are shown to match the lower bound provided in [Ouyang & Xu, 2018]. For more details on the lower and upper bounds on this formulation, the interested readers are referred to [Ouyang & Xu, 2018]. Finally, for the bilinear coupling problem (3), the authors of [Ibrahim et al., 2019] show that a lower bound of $\mathcal{O}\left(\sqrt{\frac{L_{xy}^2 + \mu_x \mu_y}{\mu_{xy}^2 + \mu_x \mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ can be derived, where μ_{xy} stands for the minimum singular value of the coupling matrix A. It is interesting that this result covers the linear convergence phenomenon for pure bilinear saddle point problem [Azizian et al., 2020] where $f(x) \equiv g(y) \equiv 0$. Another remark is that, due to the special construction of the worst-case instance and algorithm class, [Ibrahim et al., 2019] cannot characterize the impact of L_x and L_y as well as the lower bound for proximal algorithm class.

Other than studies on the first-order algorithms, there are also studies on the higher-order methods as well. For example, in [Arjevani et al., 2019] lower iteration complexity bounds for second-order methods are considered, and in [Agarwal & Hazan, 2017, Nesterov, 2018a] lower iteration complexity bounds are presented for general higher-order (tensor) methods. For smooth nonconvex optimization, in [Carmon et al., 2019] the iteration complexity lower bounds for first-order methods are considered, while in [Carmon et al., 2017] that for higher-order methods are considered.

Another line of research is for the non-conex/concave min-max saddle point problems; see [Jin et al., 2019b, Jin et al., 2019a] and the references therein. To guarantee convergence, additional structures are often needed. For example, if one assumes that the solutions of the problem satisfy the Minty variational inequality [Lin et al., 2018] then convergent algorithm can be constructed. Another important situation is when F is concave in y. In that case, convergence and iteration complexity to a stationary solution is possible; see e.g. [Lu et al., 2019]. For more literatures in this type of problems, we refer the interested readers to [Lin et al., 2019] and the references therein.

Organization. This paper is organized as follows. In Section 2, we introduce two different algorithm classes (with or without proximal-operators). In Section 3, we construct a worst-case example for problem class $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$ and derive the corresponding lower iteration complexity bound for the algorithm class allowing proximal-operators. An optimal algorithm is discussed in this case. In

Section 4, we construct the worst-case example for problem class $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$ and establish the corresponding lower complexity bound for the first-order method (without any proximal oracles). Optimal algorithms under several special parameter regimes are discussed. Finally, we conclude the paper in Section 6.

2 The first-order algorithm classes

In this section, we discuss some preliminaries for the strongly convex and strongly concave saddle point problem. Then, we shall introduce two algorithm classes to set the ground for our discussion, and we shall also note specific known algorithms as representative members in those algorithm classes.

2.1 Primal function, dual function, and the duality gap

First, we define $\Phi(\cdot)$ to be the primal function and $\Psi(\cdot)$ to be the dual function of the saddle point problem $\min_x \max_y F(x, y)$, respectively, with the following definitions

$$\Phi(x) := \max_y F(x, y) \qquad \text{and} \qquad \Psi(y) := \min_x F(x, y) \tag{5}$$

As the maximum of a class of μ_x-strongly convex function, we know $\Phi(x)$ is a μ_x-strongly convex function. Similarly, $\Psi(y)$ is a μ_y-strongly concave function. We define the duality gap as

$$\Delta(x, y) := \max_{y'} F(x, y') - \min_{x'} F(x', y) = \Phi(x) - \Psi(y)$$

Suppose the unique solution of this min-max problem is (x^*, y^*). By the strong duality theorem, we know for any x and y it holds that

$$\Phi(x) \geq \min_{x'} \Phi(x') = \Phi(x^*) = F(x^*, y^*) = \Psi(y^*) = \max_{y'} \Psi(y') \geq \Psi(y)$$

Together with the μ_x-strong convexity of Φ and the μ_y-strong concavity of Ψ, we further have

$$\Delta(x, y) = \Phi(x) - \Phi(x^*) + \Psi(y^*) - \Psi(y) \geq \frac{\mu_x}{2}\|x - x^*\|^2 + \frac{\mu_y}{2}\|y - y^*\|^2 \tag{6}$$

Now, suppose that $(\widetilde{x}_k, \widetilde{y}_k)$ is the approximate solution generated after k iterations of an algorithm. Our aim is to lower bound the distance between $(\widetilde{x}_k, \widetilde{y}_k)$ and (x^*, y^*). By (6), this would construct a lower iteration complexity bound in terms of the duality gap as well.

2.2 Proximal algorithm class

First, let us consider the bilinearly coupled problem class (3) as introduced in Definition 2:

$$\min_x \max_y F(x, y) := f(x) + x^\top A y - g(y)$$

For this special problem class, let us consider the lower iteration bound of the algorithm class where the proximal oracles (4) are available.

Definition 3 (Proximal algorithm class)**.** *In each iteration, the iterate sequence* $\{(x^k, y^k)\}_{k=0,1,\ldots}$ *are generated so that* $(x^k, y^k) \in \mathcal{H}_x^k \times \mathcal{H}_y^k$*. These subspaces are generated with* $\mathcal{H}_x^0 = \mathrm{Span}\{x^0\}, \mathcal{H}_y^0 = \mathrm{Span}\{y^0\}$ *and*

$$\begin{cases} \mathcal{H}_x^{k+1} := \mathrm{Span}\{x^i, \mathrm{prox}_{\gamma_i f}(\widehat{x}^i - \gamma_i A\widehat{y}^i) \ : \forall \widehat{x}^i \in \mathcal{H}_x^i,\ \widehat{y}^i \in \mathcal{H}_y^i,\ 0 \le i \le k\} \\ \mathcal{H}_y^{k+1} := \mathrm{Span}\{y^i, \mathrm{prox}_{\sigma_i g}(\widehat{y}^i + \sigma_i A^\top \widehat{x}^i) : \forall \widehat{x}^i \in \mathcal{H}_x^i,\ \widehat{y}^i \in \mathcal{H}_y^i,\ 0 \le i \le k\} \end{cases} \tag{7}$$

Remark that when applying the proximal oracles, it is not necessary to use the most recent iterate x^k as the proximal center. Neither is it necessary to use the gradients of the coupling term (namely the $A^\top x$ and Ay terms) at the current iterate. Instead, the algorithm class allows the usage of the combination of *any* points in the historical search space. We shall also remark that the algorithm class in Definition 3 does not necessarily need to update x and y at the same time, because setting $x^{k+1} = x^k$ or $y^{k+1} = y^k$ also satisfies Definition 3. Thus this algorithm class also includes the methods that alternatingly update x and y. Below is a sample algorithm in this class.

Example 1 (Algorithm 3 in [Chambolle & Pock, 2011])**.** Initialize with $\gamma = \frac{1}{L_{xy}}\sqrt{\frac{\mu_y}{\mu_x}}$, $\sigma = \frac{1}{L_{xy}}\sqrt{\frac{\mu_x}{\mu_y}}$, and $\theta = \frac{L_{xy}}{2\sqrt{\mu_x\mu_y}+L_{xy}}$. Set $\widetilde{x}^0 = x^0$. Then the algorithm proceeds as

$$\begin{cases} y^{k+1} = \mathrm{prox}_{\sigma g}(y^k + \sigma A^\top \widetilde{x}^k) \\ x^{k+1} = \mathrm{prox}_{\gamma f}(x^k - \gamma A y^{k+1}) \\ \widetilde{x}^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k) \end{cases} \tag{8}$$

It can be observed that this algorithm takes the alternating order of update, by slightly manipulating the index, it can be written in the form of (7) in Definition 3. The complexity of this method is $\mathcal{O}\left(\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$.

2.3 Pure first-order algorithm class

In constrast to the previous section, here we consider the more general problem class $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$:

$$\min_x \max_y F(x, y)$$

For such problems, we refer to the algorithm class as the *pure first-order methods*, meaning that there is no proximal oracle in the design of algorithms in this class.

Definition 4 (Pure first-order algorithm class)**.** *In each iteration, the sequence* $\{(x_k, y_k)\}_{k=0,1,\ldots}$ *is generated so that* $(x^k, y^k) \in \mathcal{H}_x^k \times \mathcal{H}_y^k$*, with* $\mathcal{H}_x^0 = \mathrm{Span}\{x^0\}$*,* $\mathcal{H}_y^0 = \mathrm{Span}\{y^0\}$*, and*

$$\begin{cases} \mathcal{H}_x^{k+1} := \mathrm{Span}\{x^i, \nabla_x F(\widehat{x}^i, \widehat{y}^i) : \forall \widehat{x}^i \in \mathcal{H}_x^i, \widehat{y}^i \in \mathcal{H}_y^i, 0 \le i \le k\} \\ \mathcal{H}_y^{k+1} := \mathrm{Span}\{y^i, \nabla_y F(\widehat{x}^i, \widehat{y}^i) : \forall \widehat{x}^i \in \mathcal{H}_x^i, \widehat{y}^i \in \mathcal{H}_y^i, 0 \le i \le k\} \end{cases} \tag{9}$$

Similar to our earlier comments on the proximal algorithm class, in this class of algorithms the gradients at any combination of points in the historical search space are allowed. The algorithm class also includes the methods that alternatingly update between x and y, or even the double loop algorithms that optimize one side until certain accuracy is achieved before switching to the other side. At that level of generality, it indeed accommodates many updating schemes. To illustrate this point, let us present below some sample algorithms in this class.

The first example is a double loop scheme, in which the primal function $\Phi(x)$ is optimized approximately. Specifically, let $y^*(x) = \text{argmax}_y F(x,y)$, by Danskin's theorem, $\nabla\Phi(x) = \nabla_x F(x, y^*(x))$; see e.g. [Bertsekas, 1997, Rockafellar, 1970]. Therefore, one can apply Nesterov's accelerated gradient method to minimize $\Phi(x)$. The double loop scheme performs this procedure approximately.

Example 2 (Double loop schemes, [Sanjabi et al., 2018])**.** Denote $\alpha_1 = \sqrt{\frac{\mu_x}{L_{\Phi,x}}}$ and $\alpha_2 = \sqrt{\frac{\mu_y}{L_y}}$, where $L_{\Phi,x} = L_x + \frac{L_{xy}^2}{\mu_y}$ is the Lipschitz constant of $\nabla\Phi(x)$ (see [Sanjabi et al., 2018]). Given (x^0, y^0) and define $\overline{x}^0 = x^0$, the double loop scheme works as follows:

$$\begin{cases} x^{k+1} = \overline{x}^k - \frac{1}{L_{\Phi,x}}\nabla_x F(\overline{x}^k, y^k) \\ \overline{x}^{k+1} = x^{k+1} + \frac{1-\sqrt{\alpha_2}}{1+\sqrt{\alpha_2}}(x^{k+1} - x^k) \end{cases} \qquad \text{for} \quad k = 0, 1, ..., T_1$$

where the point y^k is generated by an inner loop of accelerated gradient iterations

$$\begin{cases} w^{t+1} = \overline{w}^t + \frac{1}{L_y}\nabla_y F(\overline{x}^k, \overline{w}^t) \\ \overline{w}^{t+1} = w^{t+1} + \frac{1-\sqrt{\alpha_1}}{1+\sqrt{\alpha_1}}(w^{t+1} - w^t) \end{cases} \qquad \text{for} \quad t = 0, 1, ..., T_2 \ \text{ and } \ w^0 = \overline{w}^0 = y^{k-1}$$

Then, set $y^k := w^{T_2+1}$ to be the last iterate of the inner loop.

For simplicity, we have applied a specific scheme of acceleration [Nesterov, 2018b] which does not work for nonstrongly-convex problems. In principle, the FISTA scheme can also be used. For this scheme, with properly chosen T_1 and T_2, the iteration complexity $\mathcal{O}\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L_{xy}^2}{\mu_x\mu_y}} \cdot \sqrt{\frac{L_y}{\mu_y}} \log^2\left(\frac{1}{\epsilon}\right)\right)$ is achievable.

In the following, we also list examples of several single loop algorithms, including the gradient descent-ascent method (GDA), the extra-gradient (EG) method [Korpelevich, 1976] (a special case of mirror-prox algorithm [Nemirovski, 2004]), and the accelerated dual extrapolation (ADE) [Nesterov & Scrimali, 2006].

Example 3 (Single loop algorithms)**.** Let $L = \max\{L_x, L_y, L_{xy}\}, \mu = \min\{\mu_x, \mu_y\}$. Given the initial solution (x^0, y^0), the algorithms proceed as follows:

$$(GDA) \qquad \begin{cases} x^{k+1} = x^k - \eta_1\nabla_x F(x^k, y^k) \\ y^{k+1} = y^k + \eta_1\nabla_y F(x^k, y^k) \end{cases}$$

$$(EG) \qquad \begin{cases} \widetilde{x}^{k+1} = x^k - \eta_2\nabla_x F(x^k, y^k) \\ \widetilde{y}^{k+1} = y^k + \eta_2\nabla_y F(x^k, y^k) \end{cases} \quad \text{and} \quad \begin{cases} x^{k+1} = x^k - \eta_2\nabla_x F(\widetilde{x}^{k+1}, \widetilde{y}^{k+1}) \\ y^{k+1} = y^k + \eta_2\nabla_y F(\widetilde{x}^{k+1}, \widetilde{y}^{k+1}) \end{cases}$$

$$(ADE) \qquad \begin{cases} x^{k+1} = x^k - \eta_3\left(\frac{\mu}{L+\mu}\nabla_x F(x^k, y^k) + \frac{L}{L+\mu}\nabla_x F(\widetilde{x}^{k+1}, \widetilde{y}^{k+1})\right) \\ y^{k+1} = y^k + \eta_3\left(\frac{\mu}{L+\mu}\nabla_y F(x^k, y^k) + \frac{L}{L+\mu}\nabla_y F(\widetilde{x}^{k+1}, \widetilde{y}^{k+1})\right) \end{cases}$$

where $\eta_1 = \mathcal{O}\left(\frac{\mu}{L^2}\right)$, $\eta_2 = \mathcal{O}\left(\frac{1}{L}\right), \eta_3 = \mathcal{O}\left(\frac{1}{L}\right)$. The iterative points $(\widetilde{x}^{k+1}, \widetilde{y}^{k+1})$ in *(ADE)* are the same as that in *(EG)*, except that η_2 is replaced by η_3.

The original update of (ADE) algorithm is rather complex since it involves the handling of constraints. In the unconstrained case, it can be simplified to the current form, which is a mixture of (GDA) and (EG). The corresponding iteration complexity bounds are $\mathcal{O}\left(\frac{L^2}{\mu^2}\log\left(\frac{1}{\epsilon}\right)\right)$ for (GDA), and $\mathcal{O}\left(\frac{L}{\mu}\log\left(\frac{1}{\epsilon}\right)\right)$ for both (EG) and (ADE).

2.4 General deterministic algorithm classes without linear span structure

Although all the reviewed first-order methods satisfy the *linear span* property in the proximal algorithm class in Definition 3 and the pure first-order algorithm class in Definition 4, this does not exclude the possibility of the deriving an algorithm that does not satisfy the linear span property. Therefore, we also define the general deterministic proximal algorithm class and the general deterministic pure first-order algorithm class as follows, whose iteration complexity lower bound can be generalized from their linear span counterpart through the technique of *adversary rotation*.

Definition 5 (General proximal algorithm class)**.** *Consider the problem* (3) *in the problem class* $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$*, denote* $\theta = (L_{xy}, \mu_x, \mu_y)$ *as the corresponding problem parameters. Let algorithm* $\mathcal{A}$ *belong to the general proximal algorithm class. Then* $\mathcal{A}$ *consists of a sequence of deterministic mappings* $\{(\mathcal{A}_x^1, \mathcal{A}_y^1, \mathcal{A}_u^1, \mathcal{A}_v^1), (\mathcal{A}_x^2, \mathcal{A}_y^2, \mathcal{A}_u^2, \mathcal{A}_v^2)...\}$ *such that the iterate sequence* $\{(x^k, y^k)\}_{k=0,1,...}$ *and the output sequence* $\{(\widetilde{x}^k, \widetilde{y}^k)\}_{k=0,1,...}$ *are generated by*

$$\begin{cases} (x^k, \widetilde{x}^k) := \mathcal{A}_x^k\left(\theta; x^0, Ay^0, ..., x^{k-1}, Ay^{k-1}; \mathrm{prox}_{\gamma_k f}(u^k)\right) \\ (y^k, \widetilde{y}^k) := \mathcal{A}_y^k\left(\theta; y^0, A^\top x^0, ..., y^{k-1}, A^\top x^{k-1}; \mathrm{prox}_{\sigma_k g}(v^k)\right), \end{cases} \tag{10}$$

where $u^k = \mathcal{A}_u^k(\theta; x^0, Ay^0, ..., x^{k-1}, Ay^{k-1})$*,* $v^k = \mathcal{A}_v^k(\theta; y^0, A^\top x^0, ..., y^{k-1}, A^\top x^{k-1})$*, and* (x^0, y^0) *is any given initial solution.*

One remark is that the input of the proximal mapping $\mathrm{prox}_{\gamma_k f}(\cdot)$ is constructed with other inputs to the $\mathcal{A}_x^k$, i.e., there could be another deterministic mapping $\mathcal{A}_u^k$ to generate a vector $u^k = \mathcal{A}_u^k(\theta; x^0, Ay^0, ..., x^{k-1}, Ay^{k-1})$ and then $\mathrm{prox}_{\gamma_k f}(u^k)$ is passed to the mapping $\mathcal{A}_x^k$. This $\mathcal{A}_u^k$ does not need to be linear. The situation for v^k and $\mathcal{A}_v^k$ is similar. Similar to the general proximal algorithm class, the general pure first-order algorithm class is defined as follows.

Definition 6 (General pure first-order algorithm class)**.** *Consider the problem* (1) *in the problem class* $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$*, denote* $\theta = (L_x, L_y, L_{xy}, \mu_x, \mu_y)$ *as the corresponding problem parameters. Let algorithm* $\mathcal{A}$ *belong to the general pure first-order algorithm class. Then* $\mathcal{A}$ *consists of a sequence of deterministic mappings* $\{\mathcal{A}_x^1, \mathcal{A}_y^1, \mathcal{A}_x^2, \mathcal{A}_y^2, ...\}$ *such that the iterate sequence* $\{(x^k, y^k)\}_{k=0,1,...}$ *and the output sequence* $\{(\widetilde{x}^k, \widetilde{y}^k)\}_{k=0,1,...}$ *are generated by*

$$\begin{cases} (x^k, \widetilde{x}^k) := \mathcal{A}_x^k\left(\theta; x^0, \nabla_x F(x^0, y^0), ..., x^{k-1}, \nabla_x F(x^{k-1}, y^{k-1})\right) \\ (y^k, \widetilde{y}^k) := \mathcal{A}_y^k\left(\theta; y^0, \nabla_y F(x^0, y^0), ..., y^{k-1}, \nabla_y F(x^{k-1}, y^{k-1})\right), \end{cases} \tag{11}$$

given any initial solution (x^0, y^0)*.*

A remark is that the gradients $\nabla_x F(\cdot, \cdot)$ and $\nabla_y F(\cdot, \cdot)$ actually do not need to be taken on the previous iterates $\{(x^0, y^0), (x^1, y^1), ..., (x^{k-1}, y^{k-1})\}$. Similar to the proximal case, they can also be taken on some other $\{(u_{k-1}^0, v_{k-1}^0), (u_{k-1}^1, v_{k-1}^1), ..., (u_{k-1}^{k-1}, v_{k-1}^{k-1})\}$ that are generated by some mappings $\{(\mathcal{A}_u^{k-1,0}, \mathcal{A}_v^{k-1,0}), (\mathcal{A}_u^{k-1,1}, \mathcal{A}_v^{k-1,1}), ..., (\mathcal{A}_u^{k-1,k-1}, \mathcal{A}_v^{k-1,k-1})\}$. The reason that we do not consider this more general form is twofold. First, the simpler form in Definition 6 has already covered all the discussed algorithms. Second, this more general form actually shares the same iteration complexity lower bound, despite the technical complications involved. Therefore, in this paper we shall only include the gradients at the past iterates as the input to the algorithm.

3 Lower bound for proximal algorithms

3.1 The worst-case instance

Let us construct the following bilinearly coupled min-max saddle point problem:

$$\min_x \max_y F(x,y) := \frac{\mu_x}{2}\|x\|^2 + \frac{L_{xy}}{2}x^\top A y - \frac{\mu_y}{2}\|y\|^2 - b^\top y \tag{12}$$

where b is a vector to be determined later, and the coupling matrix A (hence A^2 and A^4) is defined as follows:

$$A = \begin{pmatrix} & & & & 1 \\ & & & 1 & -1 \\ & & 1 & -1 & \\ & \iddots & \iddots & & \\ 1 & -1 & & & \end{pmatrix},\ A^2 = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 2 \end{pmatrix},\ A^4 = \begin{pmatrix} 2 & -3 & 1 & & & & \\ -3 & 6 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 6 & -4 \\ & & & & 1 & -4 & 5 \end{pmatrix} \tag{13}$$

Note that $A^\top = A$ and $\|A\|_2 \le 2$. Therefore (12) is an instance in the problem class $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$. It is worth noting that the example (12) is the same as that in Proposition 2 of [Ibrahim et al., 2019], which is a parallel work focused on pure first-order algorithm class. Here, we use the same example to elaborate the lower bound of the proximal methods over the general bilinear coupling class $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$, as well as a warmup for the discussion of more complex problem class $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$.

Denote e_i to be the i-th unit vector, which has 1 at the i-th component and 0 elsewhere. Then by direct calculation, one can check that A^2 satisfies the following *zero-chain property* (see Chapter 2 of [Nesterov, 2018b]).

Proposition 1 (Zero-chain property)**.** *For any vector $v \in \mathbb{R}^n$, if $v \in \mathrm{Span}\{e_i : i \le k\}$ for some $1 \le k \le n-1$, then $A^2 v \in \mathrm{Span}\{e_i : i \le k+1\}$.*

This means that if v only has nonzero elements at the first k entries, then $A^2 v$ will have at most one more nonzero entry at the $(k+1)$-th position.

For problem (12), the proximal operators in (7) can be written explicitly:

$$\begin{aligned} \mathrm{prox}_{\gamma_i f}(\widehat{x}_i - \gamma_i A\widetilde{y}_i) &= \operatorname*{argmin}_x \frac{\mu_x}{2}\|x\|^2 + \frac{1}{2\gamma_i}\left\|x - (\widehat{x}_i - \frac{\gamma_i L_{xy}}{2}A\widetilde{y}_i)\right\|^2 \\ &= \frac{1}{1+\gamma_i\mu_x}\widehat{x}_i - \frac{\gamma_i L_{xy}}{2(1+\gamma_i\mu_x)}A\widetilde{y}_i \\ &\in \mathrm{Span}\{\widehat{x}_i, A\widetilde{y}_i\} \end{aligned} \tag{14}$$

Similarly, for the y block, we also have

$$\mathrm{prox}_{\sigma_i g}(\widehat{y}_i + \sigma_i A^\top \widetilde{x}_i) = \frac{\widehat{y}_i - \sigma_i b}{1+\sigma_i\mu_y} + \frac{\sigma_i L_{xy}}{2(1+\sigma_i\mu_x)}A\widetilde{x}_i \in \mathrm{Span}\{\widehat{y}_i, A\widetilde{x}_i, b\} \tag{15}$$

Let us assume the initial point to be $x^0 = y^0 = 0$ ($\mathcal{H}_x^0 = \mathcal{H}_y^0 = \{0\}$) without loss of generality. Directly substituting (14) and 15 into Definition (3) yields

$$\begin{cases} \mathcal{H}_x^1 \subseteq \mathrm{Span}\{0\} \\ \mathcal{H}_y^1 \subseteq \mathrm{Span}\{b\} \end{cases} \quad \begin{cases} \mathcal{H}_x^2 \subseteq \mathrm{Span}\{Ab\} \\ \mathcal{H}_y^2 \subseteq \mathrm{Span}\{b\} \end{cases} \quad \begin{cases} \mathcal{H}_x^3 \subseteq \mathrm{Span}\{Ab\} \\ \mathcal{H}_y^3 \subseteq \mathrm{Span}\{b, A^2b\} \end{cases} \quad \begin{cases} \mathcal{H}_x^4 \subseteq \mathrm{Span}\{Ab, A^3b\} \\ \mathcal{H}_y^4 \subseteq \mathrm{Span}\{b, A^2b\} \end{cases} \quad \cdots$$

We formally summarize this observation below:

Lemma 1. *For problem* (12)*, for any* $k \in \mathbb{N}$*, if the iterates are generated so that* $(x_k, y_k) \in \mathcal{H}_x^k \times \mathcal{H}_y^k$*, with* $\mathcal{H}_x^k$ *and* $\mathcal{H}_y^k$ *defined by* (3)*, then based on* (14) *and* (15) *the search subspaces satisfy*

$$\mathcal{H}_x^k \subseteq \begin{cases} \{0\}, & k = 1 \\ \mathrm{Span}\left\{A^{2i}(Ab) : 0 \le i \le \left\lfloor \frac{k}{2} \right\rfloor - 1\right\}, & k \ge 2 \end{cases} \quad \text{and} \quad \mathcal{H}_y^k \subseteq \mathrm{Span}\left\{A^{2i}b : 0 \le i \le \left\lceil \frac{k}{2} \right\rceil - 1\right\}$$

3.2 Lower bounding the duality gap

Let us lower bound the dual gap, which is upper bounded by the whole duality gap. To achieve this, let us first write down the dual function of problem (12) as

$$\Psi(y) = \min_x F(x, y) = -\frac{1}{2} y^\top \left(\frac{L_{xy}^2}{4\mu_x} \cdot A^2 + \mu_y \cdot I \right) y - b^\top y \tag{16}$$

For this μ_y-strongly concave dual function, we can characterize the optimal solution y^* directly by its KKT condition $\nabla\Psi(y^*) = 0$. However, the exact solution y^* does not have a simple and clear form, so we choose to characterize it by an approximate solution $\widehat{y}^*$.

Lemma 2 (Approximate optimal solution)**.** *Let us assign the value of* b *as* $b := -\frac{L_{xy}^2}{4\mu_x} e_1$*. Denote* $\alpha := \frac{4\mu_x\mu_y}{L_{xy}^2}$*, and let* $q = \frac{1}{2}\left((2+\alpha) - \sqrt{(2+\alpha)^2 - 4}\right) \in (0, 1)$ *be the smallest root of the quadratic equation* $1 - (2+\alpha)q + q^2 = 0$*. Then, an approximate optimal solution* $\widehat{y}^*$ *can be constructed as*

$$\widehat{y}_i^* = \frac{q^i}{1-q} \quad \text{for} \quad i = 1, 2, ..., n \tag{17}$$

The approximation error can be bounded by

$$\|\widehat{y}^* - y^*\| \le \frac{q^{n+1}}{\alpha(1-q)} \tag{18}$$

where $\widehat{y}_i^*$ *is the* i*-th element of* $\widehat{y}^*$*. Note that* $q < 1$ *and the lower bound is dimension-independent, hence we are free to choose* n *to make the approximation error arbitrarily small.*

Proof. First, let us substitute the value of b into the KKT system $\nabla\Psi(y^*) = 0$, by slight rearranging and scaling the terms, we get

$$\left(A^2 + \frac{4\mu_x\mu_y}{L_{xy}^2} I \right) y^* = -\frac{4\mu_x}{L_{xy}^2} b$$

Using the definition of α and b, the equation becomes

$$(A^2 + \alpha I) y^* = e_1$$

Substituting the formula of A^2 in (13), we expand the above equation as

$$\begin{cases} (1+\alpha)y_1^* - y_2^* = 1 \\ -y_1^* + (2+\alpha)y_2^* - y_3^* = 0 \\ \qquad\qquad \vdots \\ -y_{n-2}^* + (2+\alpha)y_{n-1}^* - y_n^* = 0 \\ -y_{n-1}^* + (2+\alpha)y_n^* = 0 \end{cases} \tag{19}$$

By direct calculation, we can check that $\widehat{y}^*$ satisfies the first $n-1$ equations of the KKT system (19). The last equation, however, is violated, but with a residual of size $q^{n+1}/(1-q)$. In details,

$$\begin{cases}(A^2+\alpha\cdot I)\widehat{y}^* = e_1 + \frac{q^{n+1}}{1-q}\cdot e_n\\ (A^2+\alpha\cdot I)y^* = e_1\end{cases}$$

This indicates that $\widehat{y}^* - y^* = \frac{q^{n+1}}{1-q}\cdot(A^2+\alpha I)^{-1}e_n$. Note that $\alpha^{-1}I \succeq (A^2+\alpha I)^{-1} \succ 0$, we have the approximation error bounded by (18). □

Note that in Lemma 2, we have chosen $b \propto e_1$. By the zero-chain property in Proposition 1 and Lemma 1, we can verify that the subspaces $\mathcal{H}_y^{2k-1}$ and $\mathcal{H}_y^{2k}$ satisfy

$$\mathcal{H}_y^{2k-1}, \mathcal{H}_y^{2k} \subseteq \mathrm{Span}\{b, A^2b, ..., A^{2(k-1)}b\} = \mathrm{Span}\{e_1, e_2, ..., e_k\} \tag{20}$$

This implies that for both y^{2k} and y^{2k-1}, the only possible nonzero elements are the first k ones, which again implies that the lower bound of $\|y^{2k}-y^*\|^2$ and $\|y^{2k-1}-y^*\|^2$ will be similar. For simplicity, we only discuss this lower bound for y^{2k}. The counterpart for y^{2k-1} can be obtained in a similar way. Therefore, we have the following estimations.

Lemma 3. *Assume $k \le \frac{n}{2}$ and $n \ge 2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right)$. Then*

$$\|y^{2k}-y^*\|^2 \ge \frac{q^{2k}}{16}\|y^0-y^*\|^2 \tag{21}$$

where $y^0 = 0$ is the initial solution.

Proof. By the subspace characterization (20), we have

$$\|y^{2k}-\widehat{y}^*\| \ge \sqrt{\sum_{j=k+1}^{n}(\widehat{y}_j^*)^2} = \frac{q^k}{1-q}\sqrt{q^2+q^4+\cdots+q^{2(n-k)}} \ge \frac{q^k}{\sqrt{2}}\|\widehat{y}^*\| = \frac{q^k}{\sqrt{2}}\|y^0-\widehat{y}^*\|$$

where the last inequality is due to the fact that $q<1$, $k\le\frac{n}{2}$, and $y^0=0$. If we choose n to be large enough, then $\widehat{y}^*$ and y^* can be made arbitrarily close to each other. Hence we can transform the above inequality to (21). More details of this derivation can be found in Appendix A. □

Using Lemma 3 and (6), it is then straightforward to lower bound the duality gap by

$$\Delta(x^{2k}, y^{2k}) \ge q^{2k}\cdot\frac{\mu_y\|y^*-y^0\|^2}{32}$$

Summarizing, below we present our first main result.

Theorem 1. *Let the positive parameters $\mu_x, \mu_y > 0$ and $L_{xy} > 0$ be given. For any integer k, there exists a problem instance from $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$ of form (12), with $n \ge \max\left\{2\log_q\left(\frac{\mu_x\mu_y}{\sqrt{2}L_{xy}^2}\right), 4k\right\}$, where $A \in \mathbb{R}^{n\times n}$ as defined in (13), and $b = -\frac{L_{xy}^2}{4\mu_x}e_1$. For such a problem instance, any approximate solution $(\widetilde{x}^k, \widetilde{y}^k) \in \mathcal{H}_x^k \times \mathcal{H}_y^k$ generated by the proximal algorithm class under the linear span assumption (7) satisfies*

$$\max_y F(\widetilde{x}^k, y) - \min_x F(x, \widetilde{y}^k) \ge q^k\cdot\frac{\mu_y\|y^*-y^0\|^2}{32} \quad \text{and} \quad \|\widetilde{y}^k-y^*\|^2 \ge q^k\cdot\frac{\|y^*-y^0\|^2}{16} \tag{22}$$

where $q = 1 + \frac{2\mu_x\mu_y}{L_{xy}^2} - 2\sqrt{\left(\frac{\mu_x\mu_y}{L_{xy}^2}\right)^2 + \frac{\mu_x\mu_y}{L_{xy}^2}}$.

Proposition 2. *Under the same set of assumptions of Theorem 1, if we require the duality gap to be bounded by ϵ, the number of iterations needed is at least*

$$k \geq \log\left(\frac{\mu_y\|y^* - y^0\|^2}{32\epsilon}\right) / \log(q^{-1}) = \Omega\left(\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right) \tag{23}$$

The proof of Proposition 2 is in Appendix B.

3.3 The general proximal algorithm class

Note that Theorem 1 is derived for the proximal algorithm class with the linear span assumption, in this section, we will apply the orthogonal invariance technique, introduced in [Nemirovsky, 1992], to generalize the result of Theorem 1 to the general proximal algorithm class without the linear span assumption.

Consider the bilinear problem class $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$ and the corresponding worst case problem (12) with $F(x,y) = \frac{\mu_x}{2}\|x\|^2 + \frac{L_{xy}}{2}x^\top Ay - \frac{\mu_y}{2}\|y\|^2 - b^\top y$, where A and b are defined in accordance with Theorem 1. We define the orthogonally rotated problem as

$$\min_x \max_y F_{U,V}(x,y) \coloneqq F(Ux, Vy) = \frac{\mu_x}{2}\|x\|^2 + \frac{L_{xy}}{2}x^\top U^\top AVy - \frac{\mu_y}{2}\|y\|^2 - b^\top Vy \tag{24}$$

where U, V are two orthogonal matrices. Therefore, it is clear that $F_{U,V} \in \mathcal{B}(L_{xy}, \mu_x, \mu_y)$. Let (x^*, y^*) be the saddle point of $F(x,y)$, then it is clear that the saddle point of $F_{U,V}(x,y)$ is $(\overline{x}^*, \overline{y}^*) = (U^\top x^*, V^\top y^*)$. Consequently, the lower bound for the general proximal algorithm class is characterized by the following theorem.

Theorem 2. *Let $\mathcal{A}$ be any algorithm from the general proximal algorithm class decribed in Definition 5. We assume the dimension n is sufficiently large for simplicity. For any integer k, then there exist orthogonal matrices U, V s.t. $F_{U,V} \in \mathcal{B}(L_{xy}, \mu_x, \mu_y)$, when applying $\mathcal{A}$ to $F_{U,V}$ with initial solution $(x^0, y^0) = (0,0)$, the iterates and output satisfies*

$$\{(x^0, y^0), ..., (x^k, y^k)\} \subseteq U^\top\mathcal{H}_x^{4k-1} \times V^\top\mathcal{H}_y^{4k-1} \qquad \text{and} \qquad (\widetilde{x}^k, \widetilde{y}^k) \in U^\top\mathcal{H}_x^{4k+1} \times V^\top\mathcal{H}_y^{4k+1}$$

where $\mathcal{H}_x^i, \mathcal{H}_y^i$ are defined by Lemma 1. Consequenty, by Theorem 1,

$$\|\widetilde{y}^k - V^\top y^*\|^2 \geq \frac{q^{4k+2}}{16}\|y^* - y^0\|^2$$

where q is given in Theorem 1. As a result, it takes $\Omega\left(\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ iterations to output a solution with $O(\epsilon)$ duality gap.

For the proof of this theorem, we only need to construct the orthogonal matrices U, V such that when the algorithm $\mathcal{A}$ is applied to $F_{U,V}$, the subspace inclusion argument $\{(x^0, y^0), ..., (x^k, y^k)\} \subseteq U^\top\mathcal{H}_x^{4k-1} \times V^\top\mathcal{H}_y^{4k-1}$ and $(\widetilde{x}^k, \widetilde{y}^k) \in U^\top\mathcal{H}_x^{4k+1} \times V^\top\mathcal{H}_y^{4k+1}$ holds. As a result,

$$\|\widetilde{y}^k - V^\top y^*\|^2 = \|V\widetilde{y}^k - y^*\|^2 \geq \min_{y \in \mathcal{H}_y^{4k+1}} \|y - y^*\|^2 \geq \frac{q^{4k+2}}{16}\|y^* - y^0\|^2$$

With this argument, the latter results follow directly from the discussion of Theorem 1. The proof of this theorem is presented in Appendix C.

3.4 Tightness of the bound

We claim the tightness of the derived lower bound by the following remark.

Remark (Tightness of the bound)**.** Consider the algorithm defined in Example 1, from [Chambolle & Pock, 2011, Chambolle & Pock, 2016]. The achieved upper complexity bound is $\mathcal{O}\left(\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y}}\cdot\log\left(\frac{1}{\epsilon}\right)\right)$, and it matches our lower bound. This means that our lower bound (23) is tight and the algorithm defined in Example 1 is an optimal algorithm in the proximal algorithm class in Definition 3.

4 Lower bound for pure first-order algorithms

4.1 The worst-case instance

In this section, we consider the lower complexity bound for the pure first-order method without any proximal oracle. In this case, only the gradient information can be used to construct the iterates and produce the approximate solution output. Similar as before, we still consider the bilinearly coupled problems:

$$\min_x\max_y F(x,y) := \frac{1}{2}x^\top(B_xA^2+\mu_xI)x + \frac{L_{xy}}{2}x^\top Ay - \frac{1}{2}y^\top(B_yA^2+\mu_yI)y - b^\top y \tag{25}$$

where b is a vector whose value will be determined later. The coefficients $B_x := \frac{L_x-\mu_x}{4}, B_y := \frac{L_y-\mu_y}{4}$ and the coupleing matrix A is defined by (13). Note that $\|A\|_2 \le 2$ and $\|A\|_2^2 \le 4$, we can check that problem (25) is an instance from the problem class $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$. This time the subspaces $\mathcal{H}_x^k$'s and $\mathcal{H}_y^k$'s are generated by the following gradients:

$$\begin{cases}\nabla_x F(x,y) = (B_xA^2+\mu_xI)x + \frac{L_{xy}}{2}Ay\\ \nabla_y F(x,y) = -(B_yA^2+\mu_yI)y + \frac{L_{xy}}{2}Ax - b\end{cases}$$

Following Definition 4, by letting $x^0 = y^0 = 0$ we have

$$\begin{cases}\mathcal{H}_x^1 \subseteq \mathrm{Span}\{0\}\\ \mathcal{H}_y^1 \subseteq \mathrm{Span}\{b\}\end{cases}\qquad\begin{cases}\mathcal{H}_x^2 \subseteq \mathrm{Span}\{Ab\}\\ \mathcal{H}_y^2 \subseteq \mathrm{Span}\{b, A^2b\}\end{cases}\qquad\begin{cases}\mathcal{H}_x^3 \subseteq \mathrm{Span}\{Ab, A^2(Ab)\}\\ \mathcal{H}_y^3 \subseteq \mathrm{Span}\{b, A^2b, A^4b\}\end{cases}\qquad\cdots$$

By induction, we get the general structure of these subspaces.

Lemma 4. *For problem* (25) *and for any* $k \in \mathbb{N}$*, if the iterates are generated so that* $(x_k, y_k) \in \mathcal{H}_x^k \times \mathcal{H}_y^k$*, with* $\mathcal{H}_x^k$ *and* $\mathcal{H}_y^k$ *defined by* (4)*, then we have*

$$\mathcal{H}_x^k \subseteq \begin{cases}\{0\}, & k=1\\ \mathrm{Span}\left\{A^{2i}(Ab) : 0\le i\le k-2\right\}, & k\ge 2\end{cases} \quad \text{and} \quad \mathcal{H}_y^k \subseteq \mathrm{Span}\left\{A^{2i}b : 0\le i\le k-1\right\}$$

Different from the discussion of last section, this time it is more convenient to deal with the primal function instead of the dual one. By partially maximizing over y we have

$$\Phi(x) := \max_y F(x,y) = \frac{1}{2}x^\top(B_xA^2+\mu_xI)x + \frac{L_{xy}^2}{8}\left(Ax-\frac{2b}{L_{xy}}\right)^\top(B_yA^2+\mu_yI)^{-1}\left(Ax-\frac{2b}{L_{xy}}\right)$$

which is μ_x-strongly convex. Therefore, the primal optimal solution x^* is completely characterized by the optimality condition $\nabla\Phi(x^*) = 0$. However, the solution of this system cannot be computed exactly. Instead, we shall construct an approximate solution $\widehat{x}^*$ to the exact solution x^*.

Lemma 5 (Root estimation)**.** *Consider a quartic equation*

$$1-(4+\alpha)x+(6+2\alpha+\beta)x^2-(4+\alpha)x^3+x^4=0 \tag{26}$$

where the constants are given by

$$\alpha=\frac{L_{xy}^2}{4B_xB_y}+\frac{\mu_x}{B_x}+\frac{\mu_y}{B_y}, \qquad \beta=\frac{\mu_x\mu_y}{B_xB_y} \tag{27}$$

As long as $L_x>\mu_x>0$, and $L_y>\mu_y>0$. Then the constants $0<\alpha,\beta<+\infty$ are well-defined positive real numbers. For this quartic equation, it has a real root $x=q$ satisfying

$$1-\left(\frac{1}{2}+\frac{1}{2\sqrt{2}}\sqrt{\frac{L_x}{\mu_x}+\frac{L_{xy}^2}{\mu_x\mu_y}+\frac{L_y}{\mu_y}}\right)^{-1}<q<1-\left(\frac{1}{2}+\frac{1}{2}\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y}+\frac{L_x}{\mu_x}+\frac{L_y}{\mu_y}-1}\right)^{-1} \tag{28}$$

The proof of this lemma is presented in Appendix D. With this lemma, we can construct the approximate solution $\widehat{x}^*$ as follows.

Lemma 6 (Approximate optimal solution)**.** *Let α,β be defined according to* (27)*, let q be a real root of quartic equation* (26) *satisfying* (28)*. Let us define a vector $\widehat{b}$ with elements given by*

$$\widehat{b}_1:=(2+\alpha+\beta)q-(3+\alpha)q^2+q^3, \qquad \widehat{b}_2:=q-1, \qquad \text{and} \quad \widehat{b}_k=0, \ \text{for } 3\le k\le n \tag{29}$$

and then assign $b=\frac{2B_xB_y}{L_{xy}}A^{-1}\widehat{b}$. Then an approximate solution $\widehat{x}^*$ is constructed as

$$\widehat{x}_i^*=q^i \quad \text{for} \quad i=1,2,...,n \tag{30}$$

The approximation error can be bounded by

$$\|\widehat{x}^*-x^*\|\le\frac{7+\alpha}{\beta}\cdot q^n \tag{31}$$

Note that $q<1$ and the lower bound is dimension-independent, hence we are free to choose n to make the approximation error arbitrarily small.

The proof of this lemma is parallel to that of Lemma 2, but is more involved; the detailed proof is in Appendix E.

Note that in this case, the vector $Ab\propto\widehat{b}\subset\mathrm{Span}\{e_1,e_2\}$. By the zero-chain property in Proposition 1, the subspace $\mathcal{H}_x^k$ described in Lemma 4 can be calculated by induction

$$\mathcal{H}_x^k\subset\mathrm{Span}\{e_1,e_2,...,e_k\} \quad \text{for} \quad k\ge 2 \tag{32}$$

Parallel to Lemma 3, we have the following lemma, whose proof is in Appendix F.

Lemma 7. *Assume $k\le\frac{n}{2}$ and $n\ge 2\log_q\left(\frac{\beta}{4\sqrt{2}(7+\alpha)}\right)+2$. Then*

$$\|x^k-x^*\|^2\ge\frac{q^{2k}}{16}\|x^*-x^0\|^2 \tag{33}$$

where $x^0=0$ is the initial solution.

Consequently, the duality gap is lower bounded by

$$\Delta(x^k, y^k) \geq \frac{\mu_x}{2}\|x^k - x^*\|^2 \geq q^{2k} \cdot \frac{\mu_x \|x^0 - x^*\|^2}{32}$$

Summarizing, we present our second main result in the following theorem.

Theorem 3. *Let positive parameters $\mu_x, \mu_y > 0$ and $L_x > \mu_x$, $L_y > \mu_y$, $L_{xy} > 0$ be given. For any integer k, there exists a problem instance in $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \nu_y)$ of form (25), with $n \geq \max\left\{2\log_q\left(\frac{7+\alpha}{\beta}\right), 2k\right\}$, the constants α, β as in (27), the matrix $A \in \mathbb{R}^{n\times n}$ as in (13), the vector $b = \frac{2B_xB_y}{L_{xy}}A^{-1}\widehat{b}$ where $\widehat{b}$ as in (29). For this problem, any approximate solution $(\widetilde{x}^k, \widetilde{y}^k) \in \mathcal{H}_x^k \times \mathcal{H}_y^k$ generated by first-order algorithm class (9) satisfies*

$$\max_y F(\widetilde{x}^k, y) - \min_x F(x, \widetilde{y}^k) \geq q^{2k} \cdot \frac{\mu_x\|x^* - x^0\|^2}{32} \quad \text{and} \quad \|\widetilde{x}^k - x^*\|^2 \geq q^{2k} \cdot \frac{\|x^* - x^0\|^2}{16} \tag{34}$$

where q satisfying (28) is a root of the quartic equation (26).

Remark. As a result, if we require the duality gap to be bounded by ϵ, then the number of iterations needed is at least

$$k \geq \frac{1}{2}\log\left(\frac{\mu_x\|x^* - x^0\|^2}{32\epsilon}\right) / \log(q^{-1}) = \Omega\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L_{xy}^2}{\mu_x\mu_y} + \frac{L_y}{\mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right) \tag{35}$$

4.2 The general pure first-order algorithm class

Consider the problem class $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$, we define the orthogonally rotated problem as

$$\min_x \max_y F_{U,V}(x, y) := \frac{1}{2}x^\top(B_xU^\top A^2U + \mu_x I)x + \frac{L_{xy}}{2}x^\top U^\top AVy - \frac{1}{2}y^\top(B_yV^\top A^2V + \mu_y I)y - b^\top V^\top y \tag{36}$$

where A, b, B_x, B_y are defined in accordance with Theorem 3 and U, V are two orthogonal matrices. Therefore, it is clear that $F_{U,V} \in \mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$. Let (x^*, y^*) be the saddle point of $F(x, y)$, then it is clear that the saddle point of $F_{U,V}(x, y)$ is $(\overline{x}^*, \overline{y}^*) = (U^\top\overline{x}^*, V^\top\overline{y}^*)$. Consequently, the lower bound for the general proximal algorithm class is characterized by the following theorem.

Theorem 4. *Let $\mathcal{A}$ be any algorithm from the general pure first-order algorithm class described in Definition 6. We assume the dimension n is sufficiently large for simplicity. For any integer k, then there exist orthogonal matrices U, V s.t. $F_{U,V} \in \mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$, when applying $\mathcal{A}$ to $F_{U,V}$ with initial solution $(x^0, y^0) = (0, 0)$, the iterates and output satisfies*

$$\{(x^0, y^0), ..., (x^k, y^k)\} \subseteq U^\top\mathcal{H}_x^{2k} \times V^\top\mathcal{H}_y^{2k} \quad \text{and} \quad (\widetilde{x}^k, \widetilde{y}^k) \in U^\top\mathcal{H}_x^{2k+1} \times V^\top\mathcal{H}_y^{2k+1}$$

where $\mathcal{H}_x^i, \mathcal{H}_y^i$ are defined by Lemma 4. Consequenty, by Theorem 1,

$$\|\widetilde{x}^k - U^\top x^*\|^2 \geq \frac{q^{4k+2}}{16}\|x^* - x^0\|^2$$

where q is defined in Theorem 3. As a result, it takes $\Omega\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L_{xy}^2}{\mu_x\mu_y} + \frac{L_y}{\mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$ iterations to output a solution with $O(\epsilon)$ duality gap.

The proof of this theorem is completely parallel to that of Theorem 2. We only need to construct the orthogonal matrices U, V such that $\{(x^0, y^0), ..., (x^k, y^k)\} \subseteq U^\top \mathcal{H}_x^{2k} \times V^\top \mathcal{H}_y^{2k}$ and $(\widetilde{x}^k, \widetilde{y}^k) \in U^\top \mathcal{H}_x^{2k+1} \times V^\top \mathcal{H}_y^{2k+1}$ hold, whose proof follows exactly the same proof procedure of Theorem 2. Then argue that

$$\|\widetilde{x}^k - U^\top x^*\|^2 = \|U\widetilde{x}^k - x^*\|^2 \geq \min_{x \in \mathcal{H}_x^{2k+1}} \|x - x^*\|^2 \geq \frac{q^{4k+2}}{16} \|x^* - x^0\|^2$$

The latter results follow Theorem 3 and we omit the proof.

4.3 Tightness of the bound

In this section, we discuss the tightness of this bound. Currently, to the best of our knowledge, there does not exist a pure first-order algorithm that can achieve the lower complexity bound provided in (35). Therefore, whether an optimal algorithm exists that can match this bound or the bound can be further improved remains an open problem. However, we shall see below that (35) under several special parameter regimes is indeed a tight bound.

Case 1: $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$ For this general class, define $L = \max\{L_x, L_y, L_{xy}\}$. A near optimal upper bound of

$$\mathcal{O}\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L \cdot L_{xy}}{\mu_x \mu_y} + \frac{L_y}{\mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$$

is obtained in [Wang & Li, 2020], which almost matches our lower bound.

Case 2: $\mathcal{F}(L_x, L_y, 0, \mu_x, \mu_y)$ In this case $L_{xy} = 0$, meaning that variables x and y are decoupled. Problem (1) becomes two independent convex problems with condition numbers $\frac{L_x}{\mu_x}$ and $\frac{L_y}{\mu_y}$ respectively. In this case (35) is reduced to

$$\Omega\left(\sqrt{\frac{L_x}{\mu_x}} \log\left(\frac{1}{\epsilon}\right) + \sqrt{\frac{L_y}{\mu_y}} \log\left(\frac{1}{\epsilon}\right)\right)$$

This is matched by running two independent Nesterov's accelerated gradient methods [Nesterov, 2018b].

Case 3: $\mathcal{F}(L, L, L, \mu, \mu)$ In this case $L_x = L_y = L_{xy} = L$, $\mu_x = \mu_y = \mu$. Then (35) is reduced to

$$\Omega\left(\frac{L}{\mu} \log\left(\frac{1}{\epsilon}\right)\right)$$

The extra-gradient algorithm (EG) and the accelerated dual extrapolation algorithm (ADE) introduced in Example 3 have achieved this bound; see e.g. [Nesterov & Scrimali, 2006, Mokhtari et al., 2019].

Case 4: $\mathcal{F}(L_x, \mathcal{O}(1) \cdot \mu_y, L_{xy}, \mu_x, \mu_y)$ In this case $L_y = \mathcal{O}(1) \cdot \mu_y$, meaning that one side of the problem is easy to solve. Then, (35) is reduced to

$$\Omega\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L_{xy}^2}{\mu_x \mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$$

For the double loop algorithm defined in Example 2, when we set the inner loop iteration to be $T_2 = \mathcal{O}\left(\sqrt{\frac{L_y}{\mu_y}}\log\left(\frac{1}{\epsilon}\right)\right) = \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)\right)$, and the outer loop iteration to be $T_1 = \mathcal{O}\left(\sqrt{\frac{L_{\Phi,x}}{\mu_x}}\log\left(\frac{1}{\epsilon}\right)\right) = \mathcal{O}\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L_{xy}^2}{\mu_x\mu_y}}\cdot\log\left(\frac{1}{\epsilon}\right)\right)$. Then, an upper bound of

$$T_1T_2 = \mathcal{O}\left(\sqrt{\frac{L_x}{\mu_x} + \frac{L_{xy}^2}{\mu_x\mu_y}}\cdot\log^2\left(\frac{1}{\epsilon}\right)\right)$$

can be guaranteed. It is tight up to a logarithmic factor.

Case 5: $\mathcal{F}(L, L, L, \mu_x, \mu_x)$ In this case $L_x = L_y = L_{xy} = L$. Then (35) is reduced to

$$\Omega\left(\sqrt{\frac{L^2}{\mu_x\mu_y}}\log\left(\frac{1}{\epsilon}\right)\right)$$

This bound has been achieved by [Lin et al., 2020] up to a logarithmic factor.

5 Reduction to lower bounds for convex-concave problems.

Note that in the previous sections, we consider the strongly-convex and strongly-concave problem classes $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$ and $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$, where $\mu_x > 0$ and $\mu_y > 0$. In this section, we show how our iteration complexity lower bounds provided in Theorem 1 and Theorem 3 can be reduced to the problem classes with $\mu_x = \mu_y = 0$. Similar reduction can also be done for the case where $\mu_x > 0, \mu_y = 0$, but is omitted in this paper.

5.1 Lower bound for pure first-order algorithm class

Unlike the strongly-convex and strongly-concave saddle point problems, the saddle point of the general convex-concave problem may not always exist. Therefore, we define a new problem class with bounded saddle point solution as follows.

Definition 7. *(Problem class $\mathcal{F}_0(L_x, L_y, L_{xy}, R_x, R_y)$) We say a function F belongs to the class $\mathcal{F}_0(L_x, L_y, L_{xy}, R_x, R_y)$ as long as: (i). $F \in \mathcal{F}(L_x, L_y, L_{xy}, 0, 0)$. (ii). The solution to $(x^*, y^*) = \operatorname{argmin}_x \operatorname{argmax}_y F(x, y)$ exists, and $\|x^*\| \le R_x$, $\|y^*\| \le R_y$.*

For this problem class, we have the following lower bound result, as a corollary of Theorem 4.

Corollary 1. *Consider applying the general first-order algorithm class defined by (4) to the problem class $\mathcal{F}_0(L_x, L_y, L_{xy}, R_x, R_y)$. For any $\epsilon > 0$, there exists a problem instance $F_\epsilon(x, y) \in \mathcal{F}_0(L_x, L_y, L_{xy}, R_x, R_y)$, such that*

$$\Omega\left(\sqrt{\frac{L_xR_x^2}{\epsilon}} + \frac{L_{xy}R_xR_y}{\epsilon} + \sqrt{\frac{L_yR_y^2}{\epsilon}}\right) \tag{37}$$

iterations are required to reduce the duality gap to ϵ.

Proof. We start the reduction by the following scaling argument. First, for any $\epsilon > 0$, let $\widehat{F}_\epsilon \in \mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$ be the worst-case instance described by Theorem 4. For our purpose, we choose

$$\mu_x = 64\epsilon/R_x^2 \qquad \text{and} \qquad \mu_y = 64\epsilon/R_y^2$$

Then by direct computation, we know that the following scaled problem satisfies

$$F_\epsilon(x, y) := a\widehat{F}_\epsilon(cx, dy) \in \mathcal{F}(ac^2L_x, ad^2L_y, acdL_{xy}, ac^2\mu_x, ad^2\mu_y)$$

We skip the parameter b since it is already used in the construction of the worst case instance $\widehat{F}_\epsilon$. Denote $(\widehat{x}^*, \widehat{y}^*) = \min_x \max_y \widehat{F}_\epsilon(x, y)$ and $(x^*, y^*) = \min_x \max_y F_\epsilon(x, y)$. Let us set

$$c = \frac{\|\widehat{x}^*\|}{R_x}, \quad d = \frac{\|\widehat{y}^*\|}{R_y}, \quad \text{and} \quad a = \min\{c^{-2}, d^{-2}\}$$

Then we have $x^* = \frac{R_x\widehat{x}^*}{\|\widehat{x}^*\|}$ and $y^* = \frac{R_y\widehat{y}^*}{\|\widehat{y}^*\|}$, and the Lipschitz constants of F_ϵ satisfy that $ac^2L_x \le L_x$, $ad^2L_y \le L_y$, and $acdL_{xy} \le L_{xy}$. Therefore, we know

$$F_\epsilon \in \mathcal{F}(ac^2L_x, ad^2L_y, acdL_{xy}, ac^2\mu_x, ad^2\mu_y) \cap \mathcal{F}_0(L_x, L_y, L_{xy}, R_x, R_y)$$

Note that purely scaling the variables and the function does not change the worst-case nature of this problem. In other words, F_ϵ is still the worst-case problem instance of the function class $\mathcal{F}(ac^2L_x, ad^2L_y, acdL_{xy}, ac^2\mu_x, ad^2\mu_y)$ and the lower bound of Theorem 3 is valid for this specific instance. Therefore, to get the duality gap less than or equal to ϵ, the number of iteration k is lower bounded by

$$\begin{aligned} k &\ge \Omega\left(\sqrt{\frac{ac^2L_x}{ac^2\mu_x} + \frac{a^2c^2d^2L_{xy}^2}{ac^2\mu_x \cdot ad^2\mu_y} + \frac{ad^2L_y}{ad^2\mu_y}} \cdot \log\left(\frac{ac^2\mu_x\|x^* - x^0\|^2}{32\epsilon}\right)\right) \qquad (38) \\ &\overset{(i)}{=} \Omega\left(\left(\sqrt{\frac{L_xR_x^2}{\epsilon}} + \frac{L_{xy}R_xR_y}{\epsilon} + \sqrt{\frac{L_yR_y^2}{\epsilon}}\right) \cdot \log\left(2ac^2\right)\right) \end{aligned}$$

where (i) is because $x^0 = 0$, $\|x^*\| = R_x$, and $\mu_x = 64\epsilon/R_x^2$. Therefore, as long as we can show that $\log\left(2ac^2\right) \ge \Omega(1)$, then the corollary is proved. However, since the details are rather technical, we shall provide a proof of $\log\left(2ac^2\right) \ge \log 2$ in Appendix G. □

As a remark, by setting $L_y = 0$, the lower bound (37) implies the result derived in [Ouyang & Xu, 2018]. When $L_x = L_y = L_{xy} = L$, the lower bound (37) implies the result derived in [Nemirovsky, 1992]. The reduction for the general pure first-order algorithm class defined by (6) without the linear span assumption can also be done in a similar manner and is omitted for succinctness.

5.2 Lower bound for proximal algorithm class

Like Definition 7, we define a new bilinear problem class with bounded saddle point solution as follows.

Definition 8. *(Problem class $\mathcal{B}_0(L_{xy}, R_x, R_y)$) We say a function F belongs to the function class $\mathcal{B}_0(L_{xy}, R_x, R_y)$ as long as: (i). $F \in \mathcal{B}(L_{xy}, 0, 0)$. (ii). Solution $(x^*, y^*) = \operatorname{argmin}_x \operatorname{argmax}_y F(x, y)$ exists, and $\|x^*\| \le R_x$, $\|y^*\| \le R_y$.*

For this problem class, we have the following lower bound result, as a corollary of Theorem 2.

Corollary 2. *Consider applying the proximal algorithm class defined by* (5) *to the problem class* $\mathcal{B}_0(L_{xy}, R_x, R_y)$. *For any* $\epsilon > 0$, *there exists an instance* $F_\epsilon(x, y) \in \mathcal{B}_0(L_{xy}, R_x, R_y)$, *such that*

$$\Omega\left(\frac{L_{xy} R_x R_y}{\epsilon} \right) \tag{39}$$

iterations are required to reduce the duality gap to ϵ.

Remark. The lower bound in Corollary 1 is tight. An optimal algorithm is derived in [Chambolle & Pock, 2011, Chambolle & Pock, 2016].

The reduction can be done in a similar way as in Corollary 1, but is much simpler. The details are omitted here.

6 Conclusion

In this paper, we establish tight lower iteration complexity bounds for first-order methods in solving strongly convex and strongly concave saddle point problems. We analyze both pure first-order algorithms and proximal algorithms, deriving tight lower bounds in the most general parameter regimes. Specifically, for bilinear coupling problems $\mathcal{B}(L_{xy}, \mu_x, \mu_y)$, we establish a tight lower bound for both the proximal algorithm class under the linear span assumption and the general proximal algorithm class without this assumption. Similarly, for general coupling problems $\mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$, we derive lower bounds for pure first-order algorithms with and without the linear span assumption.

In special parameter regimes, these lower bounds have been matched by corresponding optimal algorithms. However, for the most general setting of min-max optimization, the discovery of an optimal algorithm that exactly matches the lower bound remains an open question. Additionally, for the case of bilinear coupling, we demonstrate how the availability of proximal operators leads to a lower bound of $\Omega\left(\sqrt{\frac{L_{xy}^2}{\mu_x \mu_y}} \cdot \log\left(\frac{1}{\epsilon}\right)\right)$, which has been achieved by existing algorithms in the literature. By employing the orthogonal invariance technique, we extend these results to the general pure first-order and proximal algorithm classes without the linear span assumption.

Finally, our results are applicable to broader convex-concave problems. By properly scaling worst-case instances, we show that several existing lower bounds for general convex-concave problems can be deduced as special cases, underscoring the generality and robustness of our results.

References

[Abadeh et al., 2015] Abadeh, S., Esfahani, P., & Kuhn, D. (2015). Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems* (pp. 1576–1584).

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

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

[Arjovsky et al., 2017] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

[Azizian et al., 2020] Azizian, W., Scieur, D., Mitliagkas, I., Lacoste-Julien, S., & Gidel, G. (2020). Accelerating smooth games by manipulating spectral shapes. *arXiv preprint arXiv:2001.00602*.

[Bertsekas, 1997] Bertsekas, D. (1997). *Nonlinear Programming*. Athena Scientific.

[Carmon et al., 2017] Carmon, Y., Duchi, J., Hinder, O., & Sidford, A. (2017). Lower bounds for finding stationary points i. *Mathematical Programming*, (pp. 1–50).

[Carmon et al., 2019] Carmon, Y., Duchi, J., Hinder, O., & Sidford, A. (2019). Lower bounds for finding stationary points ii: First-order methods. *Mathematical Programming*.

[Chambolle & Pock, 2011] Chambolle, A. & Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1), 120–145.

[Chambolle & Pock, 2016] Chambolle, A. & Pock, T. (2016). On the ergodic convergence rates of a first-order primal-dual algorithm. *Mathematical Programming*, 159(1-2), 253–287.

[Gao & Zhang, 2017] Gao, X. & Zhang, S. (2017). First-order algorithms for convex optimization with nonseparable objective and coupled constraints. *Journal of the Operations Research Society of China*, 5(2), 131–159.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems* (pp. 2672–2680).

[Ibrahim et al., 2019] Ibrahim, A., Azizian, W., Gidel, G., & Mitliagkas, I. (2019). Linear lower bounds and conditioning of differentiable games. *arXiv preprint arXiv:1906.07300*.

[Jin et al., 2019a] Jin, C., Netrapalli, P., & Jordan, M. (2019a). Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. *arXiv preprint arXiv:1902.00618*.

[Jin et al., 2019b] Jin, C., Netrapalli, P., & Jordan, M. (2019b). What is local optimality in nonconvex-nonconcave minimax optimization? *arXiv preprint arXiv:1902.00618*.

[Juditsky et al., 2011] Juditsky, A., Nemirovski, A., & Tauvel, C. (2011). Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1), 17–58.

[Korpelevich, 1976] Korpelevich, G. (1976). The extragradient method for finding saddle points and other problems. *Matecon*, 12, 747–756.

[Lin et al., 2018] Lin, Q., Liu, M., Rafique, H., & Yang, T. (2018). Solving weakly-convex-weakly-concave saddle-point problems as weakly-monotone variational inequality. *arXiv preprint arXiv:1810.10207*.

[Lin et al., 2019] Lin, T., Jin, C., & Jordan, M. (2019). On gradient descent ascent for nonconvex-concave minimax problems. *arXiv preprint arXiv:1906.00331*.

[Lin et al., 2020] Lin, T., Jin, C., & Jordan, M. (2020). Near-optimal algorithms for minimax optimization. In *Annual Conference on Learning Theory*.

[Lu et al., 2019] Lu, S., Tsaknakis, I., Hong, M., & Chen, Y. (2019). Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *arXiv preprint arXiv:1902.08294*.

[Marcotte & Dussault, 1987] Marcotte, P. & Dussault, J.-P. (1987). A note on a globally convergent newton method for solving monotone variational inequalities. *Operations Research Letters*, 6(1), 35–42.

[Mokhtari et al., 2019] Mokhtari, A., Ozdaglar, A., & Pattathil, S. (2019). A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. *arXiv preprint arXiv:1901.08511*.

[Nemirovski, 2004] Nemirovski, A. (2004). Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1), 229–251.

[Nemirovsky, 1992] Nemirovsky, A. (1992). Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2), 153–175.

[Nemirovsky & Yudin, 1983] Nemirovsky, A. & Yudin, D. (1983). *Problem complexity and method efficiency in optimization*.

[Nesterov, 2007] Nesterov, Y. (2007). Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3), 319–344.

[Nesterov, 2018a] Nesterov, Y. (2018a). *Implementable tensor methods in unconstrained convex optimization*. Technical report, CORE Discussion Paper, 2018/05.

[Nesterov, 2018b] Nesterov, Y. (2018b). *Lectures on Convex Optimization*, volume 137. Springer.

[Nesterov & Scrimali, 2006] Nesterov, Y. & Scrimali, L. (2006). Solving strongly monotone variational and quasi-variational inequalities. *Available at SSRN 970903*.

[Nisan et al., 2007] Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. (2007). *Algorithmic Game Theory*. Cambridge University Press.

[Ouyang et al., 2015] Ouyang, Y., Chen, Y., Lan, G., & Jr., E. P. (2015). An accelerated linearized alternating direction method of multipliers. *SIAM Journal on Imaging Sciences*, 8(1), 644–681.

[Ouyang & Xu, 2018] Ouyang, Y. & Xu, Y. (2018). Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *arXiv preprint arXiv:1808.02901*.

[Rockafellar, 1970] Rockafellar, R. (1970). *Convex Analysis*. Princeton University Press.

[Sanjabi et al., 2018] Sanjabi, M., Razaviyayn, M., & Lee, J. (2018). Solving non-convex non-concave min-max games under polyak-lojasiewicz condition. *arXiv preprint arXiv:1812.02878*.

[Taji et al., 1993] Taji, K., Fukushima, M., & Ibaraki, T. (1993). A globally convergent newton method for solving strongly monotone variational inequalities. *Mathematical Programming*, 58(1-3), 369–383.

[von Neumann et al., 2007] von Neumann, J., Morgenstern, O., & Kuhn, H. (2007). *Theory of Games and Economic Behavior (commemorative edition)*. Princeton University Press.

[Wang & Li, 2020] Wang, Y. & Li, J. (2020). Improved algorithms for convex-concave minimax optimization. *arXiv preprint arXiv:2006.06359*.

[Xiao et al., 2019] Xiao, L., Yu, A., Lin, Q., & Chen, W. (2019). Dscovr: Randomized primal-dual block coordinate algorithms for asynchronous distributed optimization. *Journal of Machine Learning Research*, 20(43), 1–58.

[Xu, 2017] Xu, Y. (2017). Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM Journal on Optimization*, 27(3), 1459–1484.

A Proof of Lemma 3

By the subspace characterization (20), we have

$$\|y^{2k} - \widehat{y}^*\| \geq \sqrt{\sum_{j=k+1}^{n} (\widehat{y}_j^*)^2} = \frac{q^k}{1-q}\sqrt{q^2 + q^4 + \cdots + q^{2(n-k)}} \geq \frac{q^k}{\sqrt{2}}\|\widehat{y}^*\| = \frac{q^k}{\sqrt{2}}\|y^0 - \widehat{y}^*\|$$

where the last inequality is due to the fact that $q \leq 1, k \leq \frac{n}{2}$ and $y^0 = 0$. Note that by Lemma 2, if we require $n \geq 2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right)$, then we can guarantee that

$$\|\widehat{y}^* - y^*\| \leq \frac{q^{n+1}}{\alpha(1-q)} \leq \frac{q^{\frac{n}{2}}}{\alpha} \cdot q^k \cdot \frac{q}{(1-q)} \leq \frac{1}{4} \cdot \frac{q^k}{\sqrt{2}}\|y^0 - \widehat{y}^*\| \quad \text{for} \quad \forall 1 \leq k \leq n/2 \tag{40}$$

where the last inequality is due to $\frac{q^{\frac{n}{2}}}{\alpha} \leq \frac{1}{4\sqrt{2}}$ and $q/(1-q) \leq \|y^0 - \widehat{y}^*\|$. Therefore, we have

$$\begin{aligned}
\|y^{2k} - y^*\|^2 &\geq (\|y^{2k} - \widehat{y}^*\| - \|\widehat{y}^* - y^*\|)^2 \\
&\geq \|y^{2k} - \widehat{y}^*\|^2 - 2\|y^{2k} - \widehat{y}^*\|\|\widehat{y}^* - y^*\| \\
&\geq \min_t \left\{ t^2 - 2\|\widehat{y}^* - y^*\| t : t \geq \delta_k := \frac{q^k}{\sqrt{2}}\|y^0 - \widehat{y}^*\| \right\} \\
&= \delta_k(\delta_k - 2\|\widehat{y}^* - y^*\|) \\
&\geq \frac{1}{2}\delta_k^2 = \frac{q^{2k}}{4}\|y^0 - \widehat{y}^*\|^2
\end{aligned} \tag{41}$$

where the fourth line is due to that $d(t^2 - 2\|\widehat{y}^* - y^*\| t)/dt = 2(t - \|\widehat{y}^* - y^*\|) \geq 0$ when $t \geq \delta_k$. Hence the quadratic function is monotonically increasing in the considered interval. In addition, we also have

$$\|y^0 - y^*\| \leq \|y^0 - \widehat{y}^*\| + \|\widehat{y}^* - y^*\| \leq \|y^0 - \widehat{y}^*\| + \frac{q^n}{\alpha} \cdot \frac{q}{1-q} \leq (1 + q^n/\alpha)\|y^0 - \widehat{y}^*\| \leq 2\|y^0 - \widehat{y}^*\|$$

where the third inequality is due to that $\|y^0 - \widehat{y}^*\| \geq \widehat{y}_1^* = q/(1-q)$. For the last inequality, if $\alpha \geq 1$, then $q^n/\alpha < 1$; if $\alpha \leq 1$, then $q^n/\alpha \leq \alpha/32 \leq 1$ since $n \geq 2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right)$. Combining the above two inequalities, the desired bound (21) follows.

B Proof of Proposition 2

Here we only prove the last inequality of (23). Due to the fact that $(\log(1+z))^{-1} \geq 1/z$ for $\forall z > 0$, we know

$$\begin{aligned}
(\log(q^{-1}))^{-1} &= (\log(1 + (1-q)/q))^{-1} \geq \frac{q}{1-q} \\
&= \frac{1 + \frac{2\mu_x\mu_y}{L_{xy}^2} - 2\sqrt{\left(\frac{\mu_x\mu_y}{L_{xy}^2}\right)^2 + \frac{\mu_x\mu_y}{L_{xy}^2}}}{2\sqrt{\left(\frac{\mu_x\mu_y}{L_{xy}^2}\right)^2 + \frac{\mu_x\mu_y}{L_{xy}^2}} - \frac{2\mu_x\mu_y}{L_{xy}^2}}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\sqrt{\left(\frac{\mu_x\mu_y}{L_{xy}^2}\right)^2 + \frac{\mu_x\mu_y}{L_{xy}^2}} - \frac{\mu_x\mu_y}{L_{xy}^2}}{\frac{2\mu_x\mu_y}{L_{xy}^2}} \\
&= \frac{1}{2}\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y} + 1} - \frac{1}{2} \\
&= \Omega\left(\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y}}\right)
\end{aligned}
$$

which completes the proof.

C Proof of Theorem 2

Before proceeding the proof, let us first quote a lemma from [Ouyang & Xu, 2018].

Lemma 8 (Lemma 3.1, [Ouyang & Xu, 2018]). *Let $\mathcal{X} \subsetneqq \overline{\mathcal{X}} \subseteqq \mathbb{R}^p$ be two linear subspaces. Then for any $\overline{x} \in \mathbb{R}^p$, there exists an orthogonal matrix $\Gamma \in \mathbb{R}^{p\times p}$ s.t. $\Gamma x = x, \forall x \in \mathcal{X}$ and $\Gamma\overline{x} \in \overline{\mathcal{X}}$.*

Note that for an orthogonal matrix Γ, if $\Gamma x = x$, then we also have $\Gamma^\top x = x$. Now let us start our proof of Theorem 2.

Proof. To prove this theorem, we only need to show

$$
\{(x^0, y^0), ..., (x^k, y^k)\} \subseteq U^\top\mathcal{H}_x^{4k-1} \times V^\top\mathcal{H}_y^{4k-1} \qquad \text{and} \qquad (\widetilde{x}^k, \widetilde{y}^k) \in U^\top\mathcal{H}_x^{4k+1} \times V^\top\mathcal{H}_y^{4k+1}
$$

We separate the proof into two parts.

Part I. There exist orthogonal matrices $\widehat{U}$, $\widehat{V}$ s.t. when $\mathcal{A}$ is applied to the rotated instance $F_{\widehat{U},\widehat{V}}$, $\{(x^0, y^0), ..., (x^k, y^k)\} \subseteq \widehat{U}^\top\mathcal{H}_x^{4k-1} \times \widehat{V}^\top\mathcal{H}_y^{4k-1}$. Let $\theta = (L_{xy}, \mu_x, \mu_y)$ be the set of algorithmic parameters. To prove the result, let us construct the worst-case function $F_{U,V}$ in a recursive way.

Case $k = 1$: Let us define $U_0 = V_0 = I$. When $\mathcal{A}$ is applied to the function $F_{U_0,V_0} \in \mathcal{B}(L_{xy}, \mu_x, \mu_y)$, the iterate sequence is $(x_0^0, y_0^0) = (0, 0)$ and

$$
\begin{cases}
u_0^1 = \mathcal{A}_u^1(\theta; x_0^0, U_0^\top A V_0 y_0^0), & (x_0^1, \widetilde{x}_0^1) = \mathcal{A}_x^1(\theta; x_0^0, U_0^\top A V_0 y_0^0, \mathrm{prox}_{\gamma_1 f}(u_0^1)) \\
v_0^1 = \mathcal{A}_v^1(\theta; y_0^0, V_0^\top A U_0 x_0^0), & (y_0^1, \widetilde{y}_0^1) = \mathcal{A}_y^1(\theta; y_0^0, V_0^\top A U_0 x_0^0, \mathrm{prox}_{\sigma_1 g}(v_0^1))
\end{cases}
$$

By Lemma 8, there exists orthogonal matrices Γ_x^0 and Γ_y^0 such that $\Gamma_x^0 x_0^1 \in \mathcal{H}_x^3 = \mathrm{Span}\{Ab\}$, $\Gamma_y^0 y_0^1 \in \mathcal{H}_y^3 = \mathrm{Span}\{b, A^2 b\}$, and $\Gamma_y^0 b = (\Gamma_y^0)^\top b = b$. That is

$$
x_0^1 \in U_1^\top\mathcal{H}_x^3, \qquad \text{and} \qquad y_0^1 \in V_1^\top\mathcal{H}_y^3, \quad V_1 b = V_1^\top b = b \tag{42}
$$

where $U_1 = U_0\Gamma_x^0$ and $V_1 = V_0\Gamma_y^0$.

Now we prove that when we apply the algorithm $\mathcal{A}$ to F_{U_1,V_1}, the generated iterates $\{(x_1^0, y_1^0), (x_1^1, y_1^1)\}$ satisfy that $(x_1^0, y_1^0) = (0, 0)$ and $(x_1^1, y_1^1) = (x_0^1, y_0^1)$. That is, the first two iterates generated by $\mathcal{A}$

is completely the same for F_{U_0,V_0} and F_{U_1,V_1}. The reason is because $u_1^1 = \mathcal{A}_u^1(\theta; x_1^0, U_1^\top A V_1 y_1^0) = \mathcal{A}_u^1(\theta; 0, 0) = \mathcal{A}_u^1(\theta; x_0^0, U_0^\top A V_0 y_0^0) = u_0^1$, therefore

$$\begin{aligned}(x_1^1, \widetilde{x}_1^1) &= \mathcal{A}_x^1(\theta; x_1^0, U_1^\top A V_1 y_1^0, \mathrm{prox}_{\gamma_1 f}(u_1^1)) \\ &= \mathcal{A}_x^1(\theta; 0, 0, \mathrm{prox}_{\gamma_1 f}(u_1^1)) \\ &= \mathcal{A}_x^1(\theta; x_0^0, U_0^\top A V_0 y_0^0, \mathrm{prox}_{\gamma_1 f}(u_0^1)) \\ &= (x_0^1, \widetilde{x}_0^1)\end{aligned}$$

Through similar argument, we know $(y_1^1, \widetilde{y}_1^1) = (y_0^1, \widetilde{y}_0^1)$. Therefore, (42) indicates that

$$x_1^1 \in U_1^\top \mathcal{H}_x^3, \qquad \text{and} \qquad y_1^1 \in V_1^\top \mathcal{H}_y^3, \quad V_1 b = V_1^\top b = b \in V_1^\top \mathcal{H}_y^3 \tag{43}$$

Case $k = 2$. For the ease of the readers to follow, we perform one extra step of discussion for $k = 2$, before presenting the construction on general k.

For the problem instance F_{U_1,V_1}, the iterates generated by $\mathcal{A}$ are $(x_1^0, y_1^0) = (0, 0)$ and

$$\begin{cases} u_1^1 = \mathcal{A}_u^1(\theta; x_1^0, U_1^\top A V_1 y_1^0), & (x_1^1, \widetilde{x}_1^1) = \mathcal{A}_x^1(\theta; x_1^0, U_1^\top A V_1 y_1^0, \mathrm{prox}_{\gamma_1 f}(u_1^1)) \\ v_1^1 = \mathcal{A}_v^1(\theta; y_1^0, V_1^\top A U_1 x_1^0), & (y_1^1, \widetilde{y}_1^1) = \mathcal{A}_y^1(\theta; y_1^0, V_1^\top A U_1 x_1^0, \mathrm{prox}_{\sigma_1 g}(v_1^1)). \end{cases}$$

$$\begin{cases} u_1^2 = \mathcal{A}_u^2(\theta; x_1^0, U_1^\top A V_1 y_1^0, x_1^1, U_1^\top A V_1 y_1^1), & (x_1^2, \widetilde{x}_1^2) = \mathcal{A}_x^2(\theta; x_1^0, U_1^\top A V_1 y_1^0, x_1^1, U_1^\top A V_1 y_1^1, \mathrm{prox}_{\gamma_2 f}(u_1^2)) \\ v_1^2 = \mathcal{A}_v^2(\theta; y_1^0, V_1^\top A U_1 x_1^0, y_1^1, V_1^\top A U_1 x_1^1), & (y_1^2, \widetilde{y}_1^2) = \mathcal{A}_y^2(\theta; y_1^0, V_1^\top A U_1 x_1^0, y_1^1, V_1^\top A U_1 x_1^1, \mathrm{prox}_{\sigma_2 g}(v_1^2)). \end{cases}$$

Note that $x_1^1 \in U_1^\top \mathcal{H}_x^3 \subsetneqq U_1^\top \mathcal{H}_x^5 \subsetneqq U_1^\top \mathcal{H}_x^7$ and $\{y_1^1, b\} \subsetneqq V_1^\top \mathcal{H}_y^3 \subsetneqq V_1^\top \mathcal{H}_y^5 \subsetneqq V_1^\top \mathcal{H}_y^7$. Therefore, there exist orthogonal matrices Γ_x^1 and Γ_y^1 such that

$$\begin{cases} \Gamma_x^1 x = (\Gamma_x^1)^\top x = x, \ \forall x \in U_1^\top \mathcal{H}_x^5, \ \ \Gamma_x^1 x_1^2 \in U_1^\top \mathcal{H}_x^7 \\ \Gamma_y^1 y = (\Gamma_y^1)^\top y = y, \ \forall y \in V_1^\top \mathcal{H}_y^5, \ \ \Gamma_y^1 y_1^2 \in V_1^\top \mathcal{H}_y^7 \end{cases} \tag{44}$$

Now, let us define

$$U_2 = U_1 \Gamma_x^1 \qquad \text{and} \qquad V_2 = V_1 \Gamma_y^1$$

Now we prove that if $\mathcal{A}$ is applied to F_{U_2,V_2}, the generated iterates $\{(x_2^0, y_2^0), (x_2^1, y_2^1), (x_2^2, y_2^2)\}$ satisfy $(x_2^0, y_2^0) = (0, 0)$, $(x_2^1, y_2^1) = (x_1^1, y_1^1)$, and $(x_2^2, y_2^2) = (x_1^2, y_1^2)$. The argument for $(x_2^1, y_2^1) = (x_1^1, y_1^1)$ is almost the same as that of the case $k = 1$. We only provide the proof for $(x_2^2, y_2^2) = (x_1^2, y_1^2)$.

Next, we need to show $u_2^2 = u_1^2$, which can be proved by arguing that all the inputs to $\mathcal{A}_u^2$ are the same for both u_2^2 and u_1^2. First, it is straightforward that $x_1^0 = 0 = x_2^0, U_1^\top A V_1 y_1^0 = 0 = U_2^\top A V_2 y_2^0$. By previous argument $x_2^1 = x_1^1$. Finally, consider the last input $U_2^\top A V_2 y_2^1$, because $y_2^1 = y_1^1 \in V_1^\top \mathcal{H}_y^3 \subsetneqq V_1^\top \mathcal{H}_y^5$, we have $\Gamma_y^1 y_2^1 = y_2^1 = y_1^1 \in V_1^\top \mathcal{H}_y^3$. Then $V_2 y_2^1 = V_1 \Gamma_y^1 y_2^1 \in V_1 V_1^\top \mathcal{H}_y^3 = \mathcal{H}_y^3$. Therefore $U_1^\top A V_2 y_2^1 \in U_1^\top A \mathcal{H}_y^3 = U_1^\top \mathcal{H}_x^5$ and

$$U_2^\top A V_2 y_2^1 = \Gamma_x^1 U_1^\top A V_2 y_2^1 = U_1^\top A V_2 y_2^1 = U_1^\top A V_1 \Gamma_y^1 y_2^1 = U_1^\top A V_1 y_1^1$$

Consequently,

$$u_2^2 = \mathcal{A}_u^2(\theta; x_2^0, U_2^\top A V_2 y_2^0, x_2^1, U_2^\top A V_2 y_2^1) = \mathcal{A}_u^2(\theta; x_1^0, U_1^\top A V_1 y_1^0, x_1^1, U_1^\top A V_1 y_1^1) = u_2^1$$

and

$$(x_2^2, \widetilde{x}_2^2) \quad = \quad \mathcal{A}_x^2(\theta; x_2^0, U_2^\top A V_2 y_2^0, x_2^1, U_2^\top A V_2 y_2^1, \mathrm{prox}_{\gamma_2 f}(u_2^2))$$

$$
\begin{aligned}
&= \mathcal{A}_x^2(\theta; x_1^0, U_1^\top A V_1 y_1^0, x_1^1, U_1^\top A V_1 y_1^1, \mathrm{prox}_{\gamma_2 f}(u_1^2)) \\
&= (x_1^2, \widetilde{x}_1^2)
\end{aligned}
$$

Through a similar argument, we have $(y_2^2, \widetilde{y}_2^2) = (y_1^2, \widetilde{y}_1^2)$. By (43) and (44), we have

$$
\{x_2^0, x_2^1, x_2^2\} \in U_2^\top \mathcal{H}_x^7 \qquad \text{and} \qquad \{b, y_2^0, y_2^1, y_2^2\} \in V_2^\top \mathcal{H}_y^7 \tag{45}
$$

Case k. Suppose we already have orthogonal matrices U_{k-1}, V_{k-1}, such that when $\mathcal{A}$ is applied to $F_{U_{k-1}, V_{k-1}}$, we have

$$
\{x_{k-1}^0, x_{k-1}^1, \cdots, x_{k-1}^{k-1}\} \in U_{k-1}^\top \mathcal{H}_x^{4k-5} \qquad \text{and} \qquad \{b, y_{k-1}^0, y_{k-1}^1, \cdots, y_{k-1}^{k-1}\} \in V_{k-1}^\top \mathcal{H}_y^{4k-5} \tag{46}
$$

Again, by Lemma 8, there exist orthogonal matrices Γ_x^{k-1} and Γ_y^{k-1}, such that

$$
\begin{cases}
\Gamma_x^{k-1} x = (\Gamma_x^{k-1})^\top x = x, \ \forall x \in U_{k-1}^\top \mathcal{H}_x^{4k-3}, \ \ \Gamma_x^{k-1} x_{k-1}^k \in U_{k-1}^\top \mathcal{H}_x^{4k-1} \\
\Gamma_y^{k-1} y = (\Gamma_y^{k-1})^\top y = y, \ \forall y \in V_{k-1}^\top \mathcal{H}_y^{4k-3}, \ \ \Gamma_y^{k-1} y_{k-1}^k \in V_{k-1}^\top \mathcal{H}_y^{4k-1}
\end{cases} \tag{47}
$$

Now we define that

$$
U_k = U_{k-1} \Gamma_x^{k-1} \qquad \text{and} \qquad V_k = V_{k-1} \Gamma_y^{k-1}
$$

Therefore, similar to our previous discussion, we only need to argue that when $\mathcal{A}$ is applied to F_{U_k, V_k}, the generated iterates $\{(x_k^0, y_k^0), (x_k^1, y_k^1), \cdots, (x_k^k, y_k^k)\}$ satisfy $(x_k^i, y_k^i) = (x_{k-1}^i, y_{k-1}^i)$ for $i = 0, 1, ..., k$. We prove this argument by induction. First, it is straightforward that $(x_k^0, y_k^0) = (0, 0) = (x_{k-1}^0, y_{k-1}^0)$. Suppose $(x_k^i, y_k^i) = (x_{k-1}^i, y_{k-1}^i)$ holds for $i = 0, 1, ..., j-1 \le k-1$, now we prove $(x_k^j, y_k^j) = (x_{k-1}^j, y_{k-1}^j)$, which is almost identical to the case $k = 2$.

For any $i \in \{0, 1, ..., j-1\}$, let us show $U_{k-1}^\top A V_{k-1} y_{k-1}^i = U_k^\top A V_k y_k^i$. Because $y_k^i = y_{k-1}^i \in V_{k-1}^\top \mathcal{H}_y^{4k-5} \subsetneqq V_{k-1}^\top \mathcal{H}_y^{4k-3}$, we have $\Gamma_y^{k-1} y_k^i = y_k^i = y_{k-1}^i \in V_{k-1}^\top \mathcal{H}_y^{4k-5}$. Then $V_k y_k^i = V_{k-1} \Gamma_y^{k-1} y_k^i \in V_{k-1} V_{k-1}^\top \mathcal{H}_y^{4k-5} = \mathcal{H}_y^{4k-5}$. Therefore $U_{k-1}^\top A V_k y_k^i \in U_{k-1}^\top A \mathcal{H}_y^{4k-5} = U_{k-1}^\top \mathcal{H}_x^{4k-3}$ and

$$
U_k^\top A V_k y_k^i = (\Gamma_x^{k-1})^\top U_{k-1}^\top A V_k y_k^i = U_{k-1}^\top A V_k y_k^i = U_{k-1}^\top A V_{k-1} \Gamma_y^{k-1} y_k^i = U_{k-1}^\top A V_{k-1} y_{k-1}^i
$$

for $0 \le i \le j-1$. Consequently,

$$
\begin{aligned}
u_k^i &= \mathcal{A}_u^i(\theta; x_k^0, U_k^\top A V_k y_k^0, ..., x_k^{i-1}, U_k^\top A V_k y_k^{i-1}) \\
&= \mathcal{A}_u^i(\theta; x_{k-1}^0, U_{k-1}^\top A V_{k-1} y_{k-1}^0, ..., x_{k-1}^{i-1}, U_{k-1}^\top A V_{k-1} y_{k-1}^{i-1}) \\
&= u_{k-1}^i
\end{aligned}
$$

and

$$
\begin{aligned}
(x_k^i, \widetilde{x}_k^i) &= \mathcal{A}_x^i(\theta; x_k^0, U_k^\top A V_k y_k^0, ..., x_k^{i-1}, U_k^\top A V_k y_k^{i-1}, \mathrm{prox}_{\gamma_i f}(u_k^i)) \\
&= \mathcal{A}_x^2(\theta; x_{k-1}^0, U_{k-1}^\top A V_{k-1} y_{k-1}^0, ..., x_{k-1}^{i-1}, U_{k-1}^\top A V_{k-1} y_{k-1}^{i-1}, \mathrm{prox}_{\gamma_i f}(u_{k-1}^i)) \\
&= (x_{k-1}^i, \widetilde{x}_{k-1}^i)
\end{aligned}
$$

Through a similar argument, we have $(y_k^i, \widetilde{y}_k^i) = (y_{k-1}^i, \widetilde{y}_{k-1}^i)$. By induction, we know $(y_k^i, \widetilde{y}_k^i) = (y_{k-1}^i, \widetilde{y}_{k-1}^i)$ for $i = 0, 1, ..., k$. Consequently, we have

$$
\{x_k^0, x_k^1, \cdots, x_k^k\} \in U_k^\top \mathcal{H}_x^{4k-1} \qquad \text{and} \qquad \{b, y_k^0, y_k^1, \cdots, y_k^k\} \in V_k^\top \mathcal{H}_y^{4k-1} \tag{48}
$$

By setting $\widehat{U} = U_k$ and $\widehat{V} = V_k$, we prove the result for Part I.

Part II. There exist orthogonal matrices U, V such that when $\mathcal{A}$ is applied to the rotated instance $F_{U,V}$, $\{(x^0,y^0),...,(x^k,y^k)\} \subseteq U^\top \mathcal{H}_x^{4k-1} \times V^\top \mathcal{H}_y^{4k-1}$, and $(\widetilde{x}^k,\widetilde{y}^k) \in U^\top \mathcal{H}_x^{4k+1} \times V^\top \mathcal{H}_y^{4k+1}$.

Given the result of Part I, and let $\{(x_k^0,y_k^0),...,(x_k^k,y_k^k)\}$ and $(\widetilde{x}_k^k,\widetilde{y}_k^k)$ be generated by $\mathcal{A}$ when applied to $F_{\widehat{U},\widehat{V}} = F_{U_k,V_k}$. Therefore, by Lemma 8, there exist orthogonal matrices P, Q such that

$$\begin{cases} Px = P^\top x = x, \ \forall x \in U_k^\top \mathcal{H}_x^{4k-1}, \ \ P\widetilde{x}_k^k \in U_k^\top \mathcal{H}_x^{4k+1} \\ Qy = Q^\top y = y, \ \forall y \in V_k^\top \mathcal{H}_y^{4k-1}, \ \ Q\widetilde{y}_k^k \in V_k^\top \mathcal{H}_y^{4k+1} \end{cases} \tag{49}$$

Define $U = U_k P$, and $V = V_k Q$. Let $\{(x^0,y^0),...,(x^k,y^k)\}$ and the output $(\widetilde{x}^k,\widetilde{y}^k)$ be generated by $\mathcal{A}$ when applied to $F_{U,V}$. Then following the same line of argument of Case k, Part I, we have

$$(x^i,y^i) = (x_k^i,y_k^i), \text{ for } i = 0,1,...,k \qquad \text{and} \qquad (\widetilde{x}^k,\widetilde{y}^k) = (\widetilde{x}_k^k,\widetilde{y}_k^k)$$

Therefore, combining (49), we complete the proof of Part II. □

D Proof of Lemma 5

For the ease of analysis, let us perform a change of variable $r := (1-q)^{-1}$. Then the quartic equation (26) can be transformed to

$$f(r) := 1 + \alpha r + (\beta - \alpha) r^2 - 2\beta r^3 + \beta r^4 = 0 \tag{50}$$

Although the quartic equation does have a root formula, it is impractical to use the formula for the purpose of lower iteration complexity bound. Instead, we will provide an estimation of a large enough lower bound of r, which corresponds to lower bound on q.

First, we let $\overline{r} = \frac{1}{2} + \sqrt{\frac{\alpha}{\beta} + \frac{1}{4}}$. Then $f(\overline{r}) = 1 > 0$.

Second, we let $\underline{r} = \frac{1}{2} + \sqrt{\frac{\alpha}{2\beta} + \frac{1}{4}}$. Then,

$$\begin{aligned} f(\underline{r}) &= \beta\left(-\frac{\alpha^2}{4\beta^2} + \frac{1}{\beta}\right) \\ &= \frac{\beta}{4}\left(-\left(\frac{L_{xy}^2}{4\mu_x\mu_y} + \frac{B_x}{\mu_x} + \frac{B_y}{\mu_y}\right)^2 + \frac{4B_xB_y}{\mu_x\mu_y}\right) \\ &= \frac{\beta}{4}\left(-\left(\frac{L_{xy}^2}{4\mu_x\mu_y}\right)^2 - \frac{L_{xy}^2}{2\mu_x\mu_y}\cdot\left(\frac{B_x}{\mu_x} + \frac{B_y}{\mu_y}\right) - \left(\frac{B_x}{\mu_x} - \frac{B_y}{\mu_y}\right)^2\right) \\ &< 0 \end{aligned}$$

Together with the fact that $f(\overline{r}) = 1 > 0$, by continuity we know there is a root r between $(\underline{r},\overline{r})$, where

$$\underline{r} = \frac{1}{2} + \sqrt{\frac{\alpha}{2\beta} + \frac{1}{4}} = \frac{1}{2} + \frac{1}{2\sqrt{2}}\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y} + \frac{L_x}{\mu_x} + \frac{L_y}{\mu_y}}$$

and

$$\overline{r} = \frac{1}{2} + \sqrt{\frac{\alpha}{\beta} + \frac{1}{4}} = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y} + \frac{L_x}{\mu_x} + \frac{L_y}{\mu_y} - 1}$$

This further implies

$$1 - \underline{r}^{-1} < q < 1 - \overline{r}^{-1}$$

which proves this lemma.

E Proof of Lemma 6

First, by setting $\nabla\Phi(x^*) = 0$, we get

$$(B_xA^2 + \mu_xI)x^* + \frac{L_{xy}^2}{4}A(B_yA^2 + \mu_yI)^{-1}\left(Ax^* - \frac{2b}{L_{xy}}\right) = 0 \tag{51}$$

Note that matrix A is invertible, with

$$A^{-1} = \begin{pmatrix} & & & 1 \\ & & 1 & 1 \\ & \iddots & \iddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

Therefore, by the interchangability of $A(B_yA^2+\mu_yI) = (B_yA^2+\mu_yI)A$, we can take the inverse and get $(B_yA^2+\mu_yI)^{-1}A^{-1} = A^{-1}(B_yA^2+\mu_yI)^{-1}$. Left multiply by A and right multiply by A for both sides we get the interchangablity of

$$A(B_yA^2 + \mu_yI)^{-1} = (B_yA^2 + \mu_yI)^{-1}A$$

Applying this on equation (51) and multiplying both sides by $\frac{1}{B_xB_y}(B_yA^2+\mu_yI)$, we can equivalently write the optimality condition as

$$(A^4 + \alpha A^2 + \beta I)x^* = \widehat{b} \tag{52}$$

where

$$\alpha = \frac{L_{xy}^2}{4B_xB_y} + \frac{\mu_x}{B_x} + \frac{\mu_y}{B_y}, \qquad \beta = \frac{\mu_x\mu_y}{B_xB_y}, \quad \text{and} \quad \widehat{b} = \frac{L_{xy}}{2B_xB_y}Ab$$

The values of matrices A^2 and A^4 can be found in (13). For the ease of discussion, we may also write equation (52) in an expanded form as:

$$\begin{cases} (2+\alpha+\beta)x_1^* - (3+\alpha)x_2^* + x_3^* = \widehat{b}_1 \\ -(3+\alpha)x_1^* + (6+2\alpha+\beta)x_2^* - (4+\alpha)x_3^* + x_4^* = \widehat{b}_2 \\ x_{k-2}^* - (4+\alpha)x_{k-1}^* + (6+2\alpha+\beta)x_k^* - (4+\alpha)x_{k+1}^* + y_{k+2}^* = \widehat{b}_k \quad \text{for } 3 \le k \le n-2 \\ x_{n-3}^* - (4+\alpha)x_{n-2}^* + (6+2\alpha+\beta)x_{n-1}^* - (4+\alpha)x_n^* = \widehat{b}_{n-1} \\ x_{n-2}^* - (4+\alpha)x_{n-1}^* + (5+2\alpha+\beta)x_n^* = \widehat{b}_n \end{cases} \tag{53}$$

Because $q \in (0,1)$ is a root to the quartic equation $1-(4+\alpha)q+(6+2\alpha+\beta)q^2-(4+\alpha)q^3+q^4 = 0$, and our approximate solution $\widehat{x}^*$ is constructed as $\widehat{x}_i^* = q^i$. By direct calculation one can check that the first $n-2$ equations are satisfied and the last 2 equations are violated with controllably residuals. Indeed, for the $(n-1)$-th equation the violation is of the order q^{n+1}, and for the n-th equation the violation is of the order $|-q^n + (4+\alpha)q^{n+1} - q^{n+2}|$. Similar to the arguments for (18), we have

$$\beta\|\widehat{x}^* - x^*\| \le \|(A^4 + \alpha A^2 + \beta I)(\widehat{x}^* - x^*)\| \le (7+\alpha)q^n$$

That is, $\|\widehat{x}^* - x^*\| \le \frac{7+\alpha}{\beta}\cdot q^n$, which completes the proof.

F Proof of Lemma 7

By the subspace characterization (32), we have

$$\|x^k - \widehat{x}^*\| \geq q^k\sqrt{q^2 + \cdots + q^{2(n-k)}} \geq \frac{q^k}{\sqrt{2}}\|\widehat{x}^* - x^0\|, \quad \text{for} \quad \forall 1 \leq k \leq n/2$$

When we set $k \leq \frac{n}{2}$ and $n \geq 2\log_q\left(\frac{\beta}{4\sqrt{2}(7+\alpha)}\right) + 2$, by (31) we also have

$$\|\widehat{x}^* - x^*\| \leq q^n(7+\alpha)/\beta \leq \frac{q^k}{4\sqrt{2}}q \leq \frac{1}{4}\cdot\frac{q^k}{\sqrt{2}}\|\widehat{x}^* - x^0\|$$

Therefore, similar to (41), we also have

$$\|x^k - x^*\|^2 \geq \frac{q^{2k}}{16}\|x^* - x^0\|^2 \tag{54}$$

which proves the lemma.

G Proof of $\log(2ac^2) = \Omega(1)$

Proof. Note that $a = \min\{c^{-2}, d^{-2}\}$, if $c^{-2} \leq d^{-2}$, then $ac^2 = 1$. Consequently,

$$\log\left(2ac^2\right) = \log 2 = \Omega(1)$$

However, when $c^{-2} \geq d^{-2}$, the situation is more complicated. In this case,

$$ac^2 = \frac{c^2}{d^2} = \frac{R_y^2}{R_x^2}\cdot\frac{\|\widehat{x}^*\|^2}{\|\widehat{y}^*\|^2}$$

where $\widehat{x}^*$ and $\widehat{y}^*$ is the solution to the unscaled worst-case instance $\widehat{F}_\epsilon \in \mathcal{F}(L_x, L_y, L_{xy}, \mu_x, \mu_y)$. For the ease of discussion, let us take the dimension n is sufficiently large so that we can view the approximate solution constructed in Lemma 6 as the exact solution. Therefore, we have

$$\begin{cases}\widehat{x}^*(i) = q^i, \ i = 1, ..., n\\ (\mu_y I + B_y A^2)\widehat{y}^* = \frac{L_{xy}}{2}A\widehat{x}^* - b,\end{cases}$$

where q is defined by Theorem 3 and the second equality is due to the first-order stationary condition. Note that equation (51) also provides that

$$(B_x A^2 + \mu_x I)\widehat{x}^* + \frac{L_{xy}^2}{4}A(B_y A^2 + \mu_y I)^{-1}\left(A\widehat{x}^* - \frac{2b}{L_{xy}}\right) = 0$$

Combining the above two relations, we have

$$\begin{aligned}\widehat{y}^* &= (\mu_y I + B_y A^2)^{-1}(\frac{L_{xy}}{2}A\widehat{x}^* - b)\\ &= -\frac{2}{L_{xy}}A^{-1}(B_x A^2 + \mu_x I)\widehat{x}^*\end{aligned}$$

$$= -\frac{2B_x}{L_{xy}} A\widehat{x}^* - \frac{128\epsilon}{L_{xy}R_x^2} A^{-1}\widehat{x}^*$$

Substituting the specific forms of A and A^{-1}, we have

$$\widehat{y}^*(i) = \begin{cases} -\frac{2B_x}{L_{xy}} q^n - \frac{128\epsilon}{L_{xy}R_x^2} q^n, & i = 1 \\ -\frac{2B_x}{L_{xy}} q^{n+1-i}(1-q) - \frac{128\epsilon}{L_{xy}R_x^2} q^{n+1-i} \frac{1-q^i}{1-q}, & i \geq 2 \end{cases}$$

Therefore, we have

$$\|\widehat{y}^*\|^2 \leq \left(\frac{2B_x}{L_{xy}} + \frac{128\epsilon}{L_{xy}R_x^2}\right)^2 q^{2n} + \left(\frac{2B_x}{L_{xy}}(1-q) + \frac{128\epsilon}{L_{xy}R_x^2(1-q)}\right)^2 \sum_{i=1}^{n} q^{2i}$$

For ease of discussion, the following simplifications are made. First, we omit the q^{2n} term since $q < 1$ and n is sufficiently large. Second, note that Lemma 5 indicates that $1 - q = \Theta(\epsilon)$, the term $\frac{2B_x}{L_{xy}}(1-q) = \mathcal{O}(\epsilon)$ and the term $\frac{128\epsilon}{L_{xy}R_x^2(1-q)} = \Omega(1)$. Thus we also omit the $\frac{2B_x}{L_{xy}}(1-q)$ term which is significantly smaller. Therefore, we can write

$$\|\widehat{y}^*\|^2 \leq \left(\frac{128\epsilon}{L_{xy}R_x^2(1-q)}\right)^2 \sum_{i=1}^{n} q^{2i} = \left(\frac{128\epsilon}{L_{xy}R_x^2(1-q)}\right)^2 \|\widehat{x}^*\|^2$$

As a result,

$$ac^2 = \frac{R_y^2}{R_x^2} \cdot \frac{\|\widehat{x}^*\|^2}{\|\widehat{y}^*\|^2} \geq \frac{L_{xy}^2 R_y^2 R_x^2 (1-q)^2}{128^2\epsilon^2}$$

In Lemma 5, we also have a lower bound of $1 - q$ as

$$1 - q > \left(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{L_{xy}^2}{\mu_x\mu_y} + \frac{L_x}{\mu_x} + \frac{L_y}{\mu_y} - 1}\right)^{-1} \overset{(i)}{>} \frac{128\epsilon}{L_{xy}R_xR_y}$$

where (i) is because we have omitted the terms of smaller magnitude. Therefore,

$$\log\left(2ac^2\right) \geq \log\left(\frac{2L_{xy}^2 R_y^2 R_x^2}{128^2\epsilon^2} \cdot \frac{128^2\epsilon^2}{L_{xy}^2 R_x^2 R_y^2}\right) = \log(2) = \Omega(1)$$

Thus we complete the proof. □