Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Tensor methods for strongly convex strongly concave saddle point problems and strongly monotone variational inequalities

Petr Ostroukhov[1,3], Rinat Kamalov[1,2], Pavel Dvurechensky[3], Alexander Gasnikov[1,3,4]

submitted: March 3, 2021

[1] Moscow Institute of Physics and Technology
Institutskiy Pereulok, 9
Dolgoprudny
141701 Moscow Region
Russian Federation
E-Mail: ostroukhov@phystech.edu
kamalov.ra@phystech.edu

[2] V. A. Trapeznikov Institute of Control Sciences of RAS
Profsoyuznaya street, 65
117997 Moscow
Russian Federation

[3] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: pavel.dvurechensky@wias-berlin.de
alexander.gasnikov@wias-berlin.de

[4] Institute for Information Transmission Problems of RAS
Bolshoy Karetny per. 19, build.1
127051 Moscow
Russian Federation
E-Mail: gasnikov@yandex.ru

No. 2820
Berlin 2021

2020 *Mathematics Subject Classification.* 90C30, 90C25, 68Q25.

Key words and phrases. Variational inequality, saddle point problem, high-order smoothness, tensor methods, gradient norm minimization.

The work of P. Ostroukhov was fulfilled in Sirius (Sochi) in August 2020 and was supported by Andrei M. Raigorodskii Scholarship in Optimization. The research of A. Gasnikov was Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH$^+$ (EXC-2046/1, project ID: 390685689).

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Tensor methods for strongly convex strongly concave saddle point problems and strongly monotone variational inequalities

Petr Ostroukhov, Rinat Kamalov, Pavel Dvurechensky, Alexander Gasnikov

Abstract

In this paper we propose three p-th order tensor methods for μ-strongly-convex-strongly-concave saddle point problems (SPP). The first method is based on the assumption of p-th order smoothness of the objective and it achieves a convergence rate of $O\left(\left(\frac{L_p R^p}{\mu}\right)^{\frac{2}{p+1}} \log \frac{\mu R}{\varepsilon}\right)$, where R is an estimate of the initial distance to the solution. Under additional assumptions of first and second order smoothness of the objective we connect the first method with a locally superlinear converging algorithm and develop a second method with the complexity of $O\left(\left(\frac{L_p R^p}{\mu}\right)^{\frac{2}{p+1}} \log \frac{L_2 R \max\left\{1, \frac{L_1}{\mu}\right\}}{\mu} + \log \frac{\log \frac{L_1^3}{2\mu^2 \varepsilon}}{\log \frac{L_1 L_2}{\mu^2}}\right)$. The third method is a modified version of the second method, and it solves gradient norm minimization SPP with $\tilde{O}\left(\left(\frac{L_p R^p}{\varepsilon}\right)^{\frac{2}{p+1}}\right)$ oracle calls. Since we treat SPP as a particular case of variational inequalities, we also propose three methods for strongly monotone variational inequalities with the same complexity as the described above.

1 Introduction

In this work we focus on two types of saddle point problems (SPP). The first one is the classic minimax problem:

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} g(x, y), \tag{1}$$

where $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a convex over $\mathcal{X}$ and concave over $\mathcal{Y}$, and the sets $\mathcal{X}, \mathcal{Y}$ are convex. This is a particular case of a more general problem, called monotone variational inequality (MVI). In MVI we have a monotone operator $F : \mathcal{Z} \to \mathbb{R}^n$ over a convex set $\mathcal{Z} \subset \mathbb{R}^n$ and we need to find

$$z^* \in \mathcal{Z} : \ \forall z \in \mathcal{Z}, \langle F(z), z^* - z \rangle \leq 0. \tag{2}$$

If we set $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $F(z) = (\nabla_x g(x, y), -\nabla_y g(x, y))$, then MVI is equivalent to the min-max SPP (1).

The second problem is gradient norm minimization of SPP:

$$\min_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \|\nabla g(x, y)\|_2. \tag{3}$$

For both problems we consider unconstrained case with $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \mathbb{R}^m$. Additionally, we assume $g(x, y)$ is μ-strongly convex in $x \in \mathbb{R}^n$ and μ-strongly concave in $y \in \mathbb{R}^m$.

There is a number of papers on numerical methods for SPP (1) in convex-concave setting [13, 27, 18, 20, 28]. One of the most popular among first-order methods for this setting is the Mirror-Prox algorithm

DOI 10.20347/WIAS.PREPRINT.2820

[18], which treats saddle-point problems via solving the corresponding MVI. According to [19], this method achieves optimal complexity of $O(1/\varepsilon)$ iterations for first-order methods applied to smooth convex-concave SPP in large dimensions.

Additional assumption of strong convexity and strong concavity lead to better results. The algorithms from [25, 27, 23, 8, 15] achieve iteration complexity of $O(L/\mu \log(1/\varepsilon))$. In [14] the authors proposed an algorithm with complexity $O(L/\sqrt{\mu_x\mu_y} \log^3(1/\varepsilon))$, which matches up to a logarithmic factor the lower bound, obtained in [29]. It worths to mention that $\log^3(1/\varepsilon)$ factor can be improved, namely, it is possible to achieve iteration complexity of $O(L/\sqrt{\mu_x\mu_y} \log(1/\varepsilon))$ (see [5]).

The methods listed above use first-order oracles, and it is known from optimization that tensor methods, which use higher-order derivatives, have faster convergence rate, yet for the price of more expensive iteration. The idea of using derivatives of high order in optimization is not new (see [10]). The most common type of high-order methods use second-order oracles, for example Newton method [24, 21] and its modifications such as the cubic regularized Newton method [22]. Recently the idea of exploiting oracles beyond the second order started to attract increased attention, especially in convex optimization [1, 3, 6, 7, 4].

However, much less is known on high-order methods for SPP and MVIs. In [16] the authors propose a second-order method based on their Hybrid Proximal Extragradient framework [17]. The resulting complexity is $O(1/\varepsilon^{\frac{2}{3}})$. A recent work [2] shows how to modify Mirror-Prox method using oracles beyond second order and improves complexity to reach duality gap ε to $O(1/\varepsilon^{\frac{2}{p+1}})$ for convex-concave problems with p-th order Lipschitz derivatives. The paper [11] proposes a cubic regularized Newton method for solving SPP, which has global linear and local superlinear convergence rate if $\nabla g(x, y)$ and $\nabla^2 g(x, y)$ are Lipschitz-continuous and $g(x, y)$ is strongly convex in x and strongly concave in y.

In our work we make a next step and propose a Tensor method for strongly monotone variational inequalities and, as a corollary, a Tensor method for saddle point problems with strongly-convex-strongly-concave objective. Standing on the ideas from [2] and [11], our work can be split into three parts.

Firstly, we apply restart technique [26] to the HighOrderMirrorProx Algorithm 1 from [2], which is possible because of strong convexity and strong concavity of the objective. Such a modification improves the algorithm complexity to $O\left(\left(\frac{L_pR^p}{\mu}\right)^{\frac{2}{p+1}} \log\frac{\mu R}{\varepsilon}\right)$, where R is an upper bound for the initial distance to the solution $\|(x_1, y_1) - (x^*, y^*)\|_2$, and L_p is the Lipschitz constant of the p-th derivative.

Secondly, using an estimate of the area of local superlinear convergence, when the algorithm reaches this area, we switch to the Cubic-Regularized Newton Algorithm 3 from [11] to obtain local superlinear convergence of our algorithm. The total complexity of the final Algorithm 4 becomes $O\left(\left(\frac{L_pR^p}{\mu}\right)^{\frac{2}{p+1}} \log\frac{L_2R\max\left\{1, \frac{L_1}{\mu}\right\}}{\mu} + \log\frac{\log\frac{L_1^3}{2\mu^2\varepsilon}}{\log\frac{L_1L_2}{\mu^2}}\right)$, where L_1 and L_2 are Lipschitz constans for first and second order derivatives respectively. We want to emphasize, that the obtained $\log\log(1/\varepsilon)$ dependency on ε cannot be improved even in convex optimization [12].

Thirdly, we apply framework from [4] to the Algorithm 4 to solve the problem (3) and obtain the Algorithm 5. Its convergence rate is $\tilde{O}\left(\left(\frac{L_pR^p}{\varepsilon}\right)^{\frac{2}{p+1}}\right)$, where by tilde we mean additional multiplicative $\log$ factor.

Our paper is organized as follows. First of all, in Section 2 we provide necessary notations and assumptions (Section 2.1). Then, we present the new algorithm and obtain its convergence rate in Section 3. Firstly, in Section 3.1 we talk only about restarted algorithm from [2] and get its complexity.

Secondly, in Section 3.2 we describe how to connect it to Algorithm 3 from [11] in its quadratic convergence area and get the final Algorithm 4 convergence rate. Thirdly, in Section 3.3 we focus on how to wrap Algorithm 4 in a framework from [4] and obtain its complexity. Finally, in Section 4 we discuss our results and present some possible directions for future work.

2 Preliminaries

We use $z \in \mathbb{R}^n \times \mathbb{R}^m$ to denote the pair (x, y), $\nabla^p g(z)[h_1, ..., h_p],\ p \geq 1$ to denote directional derivative of g at z along directions $h_i \in \mathbb{R}^n \times \mathbb{R}^m, i = 1, ..., p$. The norm of the p-th order derivative is defined as

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

or equivalently

$$\|\nabla^p g(z)\|_2 := \max_{h \in \mathbb{R}^n \times \mathbb{R}^m} \{|\nabla^p g(z)[h]^p| : \|h\|_2 \leq 1\}.$$

Here we denote $\nabla^p g(z)[h, ..., h]$ as $\nabla^p g(z)[h]^p$. Also here and below $\|\cdot\|_2$ is a Euclidean norm for vectors.

Taylor approximation of some function f at point z up to the order of p we denote by

$$\Phi^f_{z,p}(\hat{z}) := \sum_{i=0}^{p} \frac{1}{i!} \nabla^i f(z)[\hat{z} - z]^i.$$

For ease of notation, the Taylor approximation of the objective g we denote by $\Phi_{(x,y),p}(\hat{x}, \hat{y}) \equiv \Phi_{z,p}(\hat{z}) \equiv \Phi^g_{z,p}(\hat{z})$.

By $D : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^n$ we denote Bregman divergence induced by a function $d : \mathcal{Z} \to \mathbb{R}$, which is continuously-differentiable and 1-strongly convex. The definition of Bregman divergence is

$$D(z_1, z_2) := d(z_1) - d(z_2) - \langle \nabla d(z_2), z_1 - z_2 \rangle.$$

In our paper we use half of squared Euclidean distance as Bregman divergence

$$D(z_1, z_2) = \frac{1}{2}\|z_1 - z_2\|_2^2. \tag{4}$$

During the analysis of convergence of our approach for gradient norm minimization (3) we will need the regularized Taylor approximation of objective g:

$$\Omega_{(x,y),p,L_p}(\hat{x}, \hat{y}) :=$$
$$\Phi_{(x,y),p}(\hat{x}, \hat{y}) + \frac{L_p(\sqrt{2})^{p-1}}{(p+1)!}\|\hat{x} - x\|_2^{p+1} - \frac{L_p(\sqrt{2})^{p-1}}{(p+1)!}\|\hat{y} - y\|_2^{p+1}.$$

Its min-max point we denote by

$$T^g_{p,L_p}(x, y) \in \text{Arg} \min_{\tilde{x} \in \mathbb{R}^n} \max_{\tilde{y} \in \mathbb{R}^m} \left\{\Omega_{(x,y),p,L_p}(\tilde{x}, \tilde{y})\right\}.$$

2.1 Assumptions

We assume objective g is strongly convex, strongly concave and p-times differentiable.

Assumption 1. $g(x, y)$ is μ-strongly convex in x and μ-strongly concave in y.

Recall that the definition of strong convexity and strong concavity is as follows.

Definition 1. $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is called μ-strongly convex and μ-strongly concave if

$$\forall x_1, x_2 \in \mathbb{R}^n,\ y \in \mathbb{R}^m \Rightarrow \langle \nabla_x g(x_1, y) - \nabla_x g(x_2, y), x_1 - x_2 \rangle \geq \mu \|x_1 - x_2\|_2^2, \tag{5}$$

$$\forall y_1, y_2 \in \mathbb{R}^m,\ x \in \mathbb{R}^n \Rightarrow \langle -\nabla_y g(x, y_1) + \nabla_y g(x, y_2), y_1 - y_2 \rangle \geq \mu \|y_1 - y_2\|_2^2. \tag{6}$$

The problem (1) is usually solved in terms of the duality gap

$$G_{\mathcal{X}\times\mathcal{Y}}(x, y) := \max_{y' \in \mathcal{Y}} g(x, y') - \min_{x' \in \mathcal{X}} g(x', y). \tag{7}$$

Since in our case $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \mathbb{R}^m$, we drop the notations of these sets from index of the duality gap and denote duality gap just as $G(x, y)$.

Before showing the connection between problem (1) and MVI (2) we need the definition of strong monotonicity.

Definition 2. $F : \mathcal{Z} \to \mathbb{R}^n$ is strongly monotone if

$$\langle F(z_1) - F(z_2), z_1 - z_2 \rangle \geq \mu \|z_1 - z_2\|_2^2. \tag{8}$$

Denote $z = \begin{pmatrix} x \\ y \end{pmatrix}$, and operator $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$:

$$F(z) = F(x, y) := \begin{pmatrix} \nabla_x g(x, y) \\ -\nabla_y g(x, y) \end{pmatrix}. \tag{9}$$

According to these definitions, the min-max problem (1) can be tackled via solving the MVI problem (2) with the specific operator F given in (9). In our work we use the following assumptions.

Assumption 2. $F(z)$ satisfies first order Lipschitz condition:

$$\begin{aligned} &\|F(z_1) - F(z_2)\|_2 \leq L_1 \|z_1 - z_2\|_2 \\ \Leftrightarrow\ &\|\nabla g(z_1) - \nabla g(z_2)\|_2 \leq L_1 \|z_1 - z_2\|_2. \end{aligned} \tag{10}$$

Assumption 3. $F(z)$ satisfies second order Lipschitz condition:

$$\begin{aligned} &\|\nabla F(z_1) - \nabla F(z_2)\|_2 \leq L_2 \|z_1 - z_2\|_2 \\ \Leftrightarrow\ &\|\nabla^2 g(z_1) - \nabla^2 g(z_2)\|_2 \leq L_2 \|z_1 - z_2\|_2. \end{aligned} \tag{11}$$

Assumption 4. $F(z)$ satisfies p-th order Lipschitz condition (p-smooth):

$$\begin{aligned} &\|\nabla^{p-1} F(z_1) - \nabla^{p-1} F(z_2)\|_2 \leq L_p \|z_1 - z_2\|_2 \\ \Leftrightarrow\ &\|\nabla^p g(z_1) - \nabla^p g(z_2)\|_2 \leq L_2 \|z_1 - z_2\|_2. \end{aligned} \tag{12}$$

We should note, that, to be consistent with [2], we define p-th order smoothness (Lipschitzness) of F as a property of $(p-1)$-th derivative of F, and, therefore, as a property of p-th derivative of g.

3 Main results

Firstly, in this section we propose the algorithm for finding ε-approximate solution to problem (1), where $F\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$ is p-smooth and μ-strongly monotone operator (assumptions 4 and 1), which allows to achieve iteration complexity of $O\left(\left(\frac{L_p R^p}{\mu}\right)^{\frac{2}{p+1}} \log\frac{\mu R}{\varepsilon}\right)$, where $R \geqslant \|z_1 - z^*\|_2$. This algorithm is a restarted modification of Algorithm 1.

Secondly, we develop the algorithm for tackling the same problem, where F is first, second and p-th order Lipschitz and μ-strongly monotone operator (all assumptions 1, 2, 3, 4). It involves the idea of exploiting previous algorithm and then switching to the Algorithm 3 in its quadratic convergence area. Thus, we obtain the Algorithm 4, that allows to achieve iteration complexity of
$O\left(\left(\frac{L_p R^p}{\mu}\right)^{\frac{2}{p+1}} \log\frac{L_2 R \max\left\{1, \frac{L_1}{\mu}\right\}}{\mu} + \log\frac{\log\frac{L_1^3}{2\mu^2\varepsilon}}{\log\frac{L_1 L_2}{\mu^2}}\right)$.

Thirdly, we propose the algorithm to find ε-approximate solution to problem (3), where all the assumptions 1, 2, 3, 4 hold. To achieve this we use the Algorithm 4, which we mentioned earliear, inside the framework from [4]. Final complexity of such algorithm in terms of norm of the gradient is $\tilde{O}\left(\left(\frac{L_p R^p}{\varepsilon}\right)^{\frac{2}{p+1}}\right)$, where by tilde we mean additional multiplicative $\log$ factor.

3.1 Restarted HighOrderMirrorProx

As mentioned earlier, in this subsection we provide restarted modification of Algorithm 1. But, initially, we need to give some additional information from [2].

Since our goal is an approximate solution to MVI, we define its ε-approximate solution as

$$z^* \in \mathcal{Z} : \forall z \in \mathcal{Z} \Rightarrow \langle F(z), z^* - z\rangle \leq \varepsilon. \tag{13}$$

At the same time, the bounds of Algorithm 1 is of the form

$$\forall z \in \mathcal{Z} \Rightarrow \frac{1}{\Gamma_T}\sum_{t=1}^{T} \gamma_t \langle F(z_t), z_t - z\rangle \leq \varepsilon, \tag{14}$$

where points z_t and $\gamma_t > 0$ are produced by the Algorithm 1, and $\Gamma_T = \sum_{t=1}^{T}\gamma_t$. The following lemma establishes the relation between (13) and (14).

Lemma 3.1 (Lemma 2.7 from [2])**.** *Let $F : \mathcal{Z} \to \mathbb{R}^n$, be monotone, $z_t \in \mathcal{Z},\ t = 1, ..., T$, and let $\gamma_t > 0$. Let $\bar{z}_t = \frac{1}{\Gamma_T}\sum_{t=1}^{T}\gamma_t z_t$. Assume* (14) *holds. Then $\bar{z}_t$ is an ε-approximate solution to* (2)*.*

MVI problem (2), which is sometimes called "weak MVI", is closely connected to strong MVI problem, where we need to find

$$z^* \in \mathcal{Z} : \forall z \in \mathcal{Z} \Rightarrow \langle F(z^*), z^* - z\rangle \leq 0. \tag{15}$$

If F is continuous and monotone, the problems (2) and (15) are eqiuvalent.

The convergence rate of the Algorithm 1 is stated in the following lemma.

Algorithm 1 HighOrderMirrorProx [Algorithm 1 in [2]]

1: **Input** $z_1 \in \mathcal{Z}, p \geq 1, T > 0.$
2: **for** t = 1 **to** T **do**
3: Determine γ_t, $\hat{z}_t$ such that:

$$\hat{z}_t = \arg\min_{z \in \mathcal{Z}} \{\gamma_t \langle \Phi^F_{z_t,p}(\hat{z}_t), z - z_t \rangle + D(z, z_t)\},$$
$$\frac{p!}{32 L_p \|\hat{z}_t - z_t\|_2^{p-1}} \leq \gamma_t \leq \frac{p!}{16 L_p \|\hat{z}_t - z_t\|_2^{p-1}},$$
$$z_{t+1} = \arg\min_{z \in \mathcal{Z}} \{\langle \gamma_t F(\hat{z}_t), z - \hat{z}_t \rangle + D(z, z_t)\}.$$

4: **end for**
5: Define $\Gamma_T \overset{def}{=} \sum_{t=1}^T \gamma_t$
6: **return** $\bar{z}_T \overset{def}{=} \frac{1}{\Gamma_T} \sum_{t=1}^T \gamma_t \hat{z}_t$.

Algorithm 2 Restarted HighOrderMirrorProx

1: **Input** $z_1 \in \mathcal{Z}, p \geq 1, 0 < \varepsilon < 1, R : R \geq \|z_1 - z^*\|_2$.
2: $k = 1$
3: $\tilde{z}_1 = z_1$
4: **for** $i \in [n]$, where $n = \left\lceil \log \frac{\mu R}{\varepsilon} + \frac{p-1}{2} \right\rceil$ **do**
5: Set $R_i = \frac{R}{2^{i-1}}$
6: Set $T_i = \left\lfloor \frac{R_i^2}{2} \left(\frac{64 L_p}{p! \mu R_i} \right)^{\frac{2}{p+1}} \right\rfloor$
7: Run Algorithm 1 with $\tilde{z}_i$, p, T_i as input
8: $\tilde{z}_{i+1} = \bar{z}_{T_i}$
9: **end for**
10: **return** $\tilde{z}_i$

Lemma 3.2 (Lemma 4.1 from [2])**.** *Suppose $F : \mathcal{Z} \to \mathbb{R}^n$ is p-th order Lipschitz and let $\Gamma_T = \sum_{t=1}^T \gamma_t$. Then, the iterates $\{\hat{z}_t\}_{t \in [T]}$, generated by Algorithm 1, satisfy*

$$\forall z \in \mathcal{Z} \Rightarrow \frac{1}{\Gamma_T} \sum_{t=1}^T \langle \gamma_t F(\hat{z}_t), \hat{z}_t - z \rangle \leq \frac{16 L_p}{p!} \left(\frac{D(z, z_1)}{T} \right)^{\frac{p+1}{2}}. \tag{16}$$

Thus, these two lemmas tell us, that if z_t and γ_t are generated by the Algorithm 1, and the right hand side of (16) is smaller than ε, then $\bar{z}_t = \frac{1}{\Gamma_T} \sum_{t=1}^T \gamma_t z_t$ is an ε-solution to regular MVI (13). Hence, it is also a solution to a convex-concave SPP. The natural way to improve the method for convex-concave problem in tighter strongly-convex-strongly-concave setting is to use restarts [26]. As a result, we obtain Algorithm 2.

Theorem 3.3. *Suppose $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$, that is defined in* (9)*, is p-th order Lipschitz and μ-strongly monotone (Assumptions 1 and 4 hold). Denote R such that $R \geq \|z_1 - z^*\|_2$. Then Algorithm 2 complexity is*

$$O\left(\left(\frac{L_p R^p}{\mu} \right)^{\frac{2}{p+1}} \log \frac{\mu R}{\varepsilon} \right). \tag{17}$$

Proof. From (15) and (16) we get the following:

$$\sum_{t=1}^{T} \gamma_t \langle F(\hat{z}_t) - F(z^*); \hat{z}_t - z^* \rangle \leq \frac{16 L_p}{p!} \left(\frac{\|z_1 - z^*\|_2^2}{2T} \right)^{\frac{p+1}{2}}. \tag{18}$$

From this and the fact that $F(x)$ is μ-strongly monotone we have

$$\begin{aligned} \mu \|\bar{z}_T - z^*\|_2^2 \overset{(*)}{\leq} \frac{\mu}{\Gamma_T} \sum_{t=1}^{T} \gamma_t \|\hat{z}_t - z^*\|_2^2 &\overset{(8)}{\leq} \frac{1}{\Gamma_T} \sum_{t=1}^{T} \gamma_t \langle F(\hat{z}_t) - F(z^*); \hat{z}_t - z^* \rangle \\ &\overset{(18)}{\leq} \frac{16 L_p}{p!} \left(\frac{\|z_1 - z^*\|_2^2}{2T} \right)^{\frac{p+1}{2}}, \end{aligned} \tag{19}$$

where (*) follows from convexity of $\|z\|_2^2$.

Now we restart the method every time the distance to solution decreases at least twice. Let T_i be such that $\|\bar{z}_{T_i} - z^*\|_2 \leq \frac{\|\tilde{z}_i - z^*\|_2}{2}$, where $\tilde{z}_i$ is the point, where we restart our algorithm. Denote $R_1 = R \geq \|\tilde{z}_1 - z^*\|_2$, $R_i = R_1/2^{i-1} \geq \|\tilde{z}_i - z^*\|_2$. Then the number of iterations before $(i+1)$-th restart is

$$\begin{aligned} \mu \|\bar{z}_{T_i} - z^*\|_2^2 &\overset{(19)}{\leq} \frac{16 L_p}{p!} \left(\frac{\|\tilde{z}_i - z^*\|_2^2}{2T_i} \right)^{\frac{p+1}{2}} \leq \frac{16 L_p}{p!} \left(\frac{R_i^2}{2T_i} \right)^{\frac{p+1}{2}} \\ &\Leftrightarrow T_i \leq \frac{R_i^2}{2} \left(\frac{64 L_p}{p! \mu R_i} \right)^{\frac{2}{p+1}} \Leftrightarrow T_i = \left\lfloor \frac{R_i^2}{2} \left(\frac{64 L_p}{p! \mu R_i} \right)^{\frac{2}{p+1}} \right\rfloor. \end{aligned}$$

Next we need to obtain the number of restarts, required to achieve the desired accuracy. Since $T_n \geq 1$, then $2T_n \geq T_n + 1 \Leftrightarrow \frac{1}{2T_n} \leq \frac{1}{T_n + 1}$. And from the definition of T_n

$$2T_n \geq T_n + 1 \geq \frac{R_n^2}{2} \left(\frac{64 L_p}{p! \mu R_n} \right)^{\frac{2}{p+1}} \Leftrightarrow \frac{1}{2T_n} \leq \frac{2}{\left(\frac{64 L_p R_n^p}{p! \mu} \right)^{\frac{2}{p+1}}}$$

So, from this fact and (16) we get

$$\begin{aligned} \frac{1}{\Gamma_{T_n}} \sum_{t=1}^{T_n} \gamma_t \langle F(\hat{z}_t) - F(z^*); \hat{z}_t - z^* \rangle &\leq \frac{16 L_p}{p!} \left(\frac{\|\tilde{z}_n - z^*\|_2^2}{2T_n} \right)^{\frac{p+1}{2}} \\ &\leq \frac{16 L_p R_n^{p+1}}{p!} \left(\frac{2}{\left(\frac{64 L_p R_n^p}{p! \mu} \right)^{\frac{2}{p+1}}} \right)^{\frac{p+1}{2}} \\ &= \frac{2^{\frac{p-3}{2}} \mu R}{2^{n-1}} \leq \varepsilon. \end{aligned}$$

$$\Leftrightarrow n \geq \log \frac{\mu R}{\varepsilon} + \frac{p-1}{2} \Leftrightarrow n = \left\lceil \log \frac{\mu R}{\varepsilon} + \frac{p-1}{2} \right\rceil$$

Algorithm 3 CRN-SPP [Algorithm 1 in [11]]

Input z_0, ε, $\bar{\gamma} > 0$, $\rho, \alpha \in (0,1)$, g satisfies Assumptions 1, 2 and 3.
while $m(z_k) > \varepsilon$ **do**
  $\gamma_k = \bar{\gamma}$
  while True **do**
    Solve the subproblem $(\tilde{x}_{k+1}, \tilde{y}_{k+1}) = \arg\min_x \max_y g_k(x, y; \gamma_k)$
    if $\gamma_k(\|\tilde{x}_{k+1} - x_k\| + \|\tilde{y}_{k+1} - y_k\|) > \mu$ **then**
      $\gamma_k = \rho\gamma_k$
    else
      break
    end if
  end while
  $d_k = (\tilde{x}_{k+1} - x_k; \tilde{y}_{k+1} - y_k)$
  if $m(z_k + \alpha d_k) < m(z_k + d_k)$ **then**
    $z_{k+1} = z_k + \alpha d_k$
  else if $m(z_k + \alpha d_k) \geq m(z_k + d_k)$ **then**
    $z_{k+1} = z_k + d_k$
  end if
  $k = k + 1$
end while
return z_k

Finally, the total number of iterations is

$$
\begin{aligned}
N &= \sum_{i=1}^{n} T_i = \sum_{i=1}^{n} \left\lceil \frac{1}{2} \left(\frac{64 L_p R_i^p}{p! \mu} \right)^{\frac{2}{p+1}} \right\rceil = \frac{1}{2} \left(\frac{64 L_p}{p! \mu} \right)^{\frac{2}{p+1}} \sum_{i=1}^{n} R_i^{\frac{2p}{p+1}} \\
&\leq \left(\frac{64 L_p}{p! \mu} \right)^{\frac{2}{p+1}} R^{\frac{2p}{p+1}} n = \left(\frac{64 L_p R^p}{p! \mu} \right)^{\frac{2}{p+1}} \left\lceil \log \frac{\mu R}{\varepsilon} + \frac{p-1}{2} \right\rceil \\
&= O\left(\left(\frac{L_p R^p}{\mu} \right)^{\frac{2}{p+1}} \log \frac{\mu R}{\varepsilon} \right).
\end{aligned}
$$

This completes the proof. □

3.2 Local quadratic convergence

Just like in previous subsection, becides introducing the Algorithm 3 and its convergence rate we need to provide some prerequisite information from [11].

Because of strong convexity and strong concavity of $g(x, y)$ a unique solution z^* to a SPP (1) exists, and $F(z^*) = 0$. Thus, we can use the following merit function from [11] during analysis of Algorithm 3 complexity.

$$
m(z) := \frac{1}{2} \|F(z)\|_2^2 = \frac{1}{2} (\|\nabla_x g(x, y)\|_2^2 + \|\nabla_y g(x, y)\|_2^2). \tag{20}
$$

Algorithm 3 solves additional saddle point subproblem on each step, that we denote as

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} g_k(x,y,\gamma_k) :=$$
$$g(z_k) + \langle \nabla g(z_k), z - z_k\rangle + \frac{1}{2}\nabla^2 g(z_k)[z-z_k]^2 + \frac{\gamma_k}{3}\|x-x_k\|_2^3 - \frac{\gamma_k}{3}\|y-y_k\|_2^3,$$

where γ_k is some constant.

This proposition provides the relation between the merit function $m(z)$ and the duality gap under assumptions 1 and 2.

Proposition 3.4 (Proposition 2.5 from [11]). *Let assumptions 1 and 2 hold. For problem* (1) *and any point* $z=(x,y)$ *the duality gap* (7) *and the merit function* (20) *satisfy the following inequalities*

$$\frac{\mu}{L_1^2}m(z) \le G(x,y) \le \frac{L_1}{\mu^2}m(z). \tag{21}$$

The next theorem proves local quadratic convergence of the Algorithm 3, and it is based on Theorem 3.6 from [11].

Theorem 3.5 (Theorem 3.6 from [11]). *Suppose* $F:\mathcal{Z}\to\mathbb{R}^n$ *is* μ*-strongly monotone, first and second order Lipschitz operator (assumptions 1, 2 and 3 hold). Let* $\{z_k\}$ *be generated by Algorithm 3 with* $\bar\gamma = \frac{L_2\mu^2}{2L^2}$, $\xi = \max\left\{1, \frac{L_1}{\mu}\right\}$ *and*

$$z_0 : \|z_0 - z^*\|_2 \le \frac{\mu}{L_2\xi}. \tag{22}$$

Then

$$\forall k\ge 0\ \|z_{k+1}-z^*\|_2 \le \frac{L_2\xi}{\mu}\|z_k - z^*\|_2^2, \tag{23}$$

Proof. Here we provide only the modified part of its proof. The rest of it can be found in [11].

If $z_{k+1} = \tilde z_{k+1} = z_k + d_k$, then

$$\|z_{k+1}-z^*\|_2 = \|\tilde z_{k+1}-z^*\|_2 \le \frac{L_2}{\mu}\|z^k - z^*\|_2^2 \le \frac{L_2\xi}{\mu}\|z_k-z^*\|_2^2.$$

Else if $z_{k+1} = \hat z_{k+1} = z_k + \alpha d_k$, then

$$\|z_{k+1}-z^*\|_2 = \|\hat z_{k+1}-z^*\|_2 \le \frac{L_1L_2}{\mu^2}\|z^k - z^*\|_2^2 \le \frac{L_2\xi}{\mu}\|z^k-z^*\|_2^2.$$

Hence, we get (23).

Now we need to find the area, where (23) works:

$$\exists c : \forall k\ge 0 : \|z_k - z^*\|_2 \le c \Rightarrow \|z_{k+1}-z^*\|_2 \le \frac{L_2\xi}{\mu}\|z_k-z^*\|_2^2$$
$$\Leftrightarrow \|z_{k+1}-z^*\|_2 \le \frac{L_2\xi}{\mu}\|z_k-z^*\|_2 \le \frac{L_2\xi c^2}{\mu} = c$$
$$\Leftrightarrow c = \frac{\mu}{L_2\xi}.$$

Thus, we get (22). □

Algorithm 4 Restarted HighOrderMirrorProx with local quadratic convergence

Input $z_1 \in \mathcal{Z}, p \geq 1, 0 < \varepsilon < 1, R : R \geq \|z_1 - z^*\|_2$, $\rho \in (0,1)$, $\alpha \in (0,1)$.
$\tilde{z}_1 = z_1$
for $i \in [n]$, where $n = \left\lceil \log \frac{L_2 R \xi}{\mu} + 1 \right\rceil$ **do**
  Set $R_i = \frac{R}{2^{i-1}}$
  Set $T_i = \left\lfloor \frac{R_i^2}{2} \left(\frac{64 L_p}{p! \mu R_i} \right)^{\frac{2}{p+1}} \right\rfloor$
  Run Algorithm 1 with $\tilde{z}_i$, p, T_i as input
  $\tilde{z}_{i+1} = \bar{z}_{T_i}$
end for
Run Algorithm 3 with $\tilde{z}_{i+1}$, $\tilde{\varepsilon} = \frac{\mu^2 \varepsilon}{L}$, $\bar{\gamma} = \frac{L_2 \mu^2}{2 L_1^2}$, ρ, α, g as input
return z_k

Our idea is to use Algorithm 2 until it reaches the area (22) and then switch to Algorithm 3. Algorithm 4 provides the pseudocode of this idea. From Proposition 3.4, our Theorem 3.3 and Theorem 3.5, we obtain the complexity of Algorithm 4.

Theorem 3.6. *Suppose $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$, that is defined in* (9)*, is μ-strongly monotone, first, second and p-th order Lipschitz operator (all assumptions 1, 2, 3, 4 hold). Denote $R : R \geq \|z_1 - z^*\|_2$ and $\xi = \max\left\{1, \frac{L_1}{\mu}\right\}$. Then the complexity of Algorithm 4 is*

$$O\left(\left(\frac{L_p R^p}{\mu} \right)^{\frac{2}{p+1}} \log \frac{L_2 \xi R}{\mu} + \log \frac{\log \frac{L_1^3}{2\mu^2 \varepsilon}}{\log \frac{L_1 L_2}{\mu^2}} \right). \tag{24}$$

Proof. First of all, we need to find the number of restarts n of Algorithm 2 to reach the area of local quadratic convergence of Algorithm 3 from (22): $\|\tilde{z}_n - z^*\|_2 \leq \frac{\mu}{L_2 \xi}$. We can choose such n, that

$$\|\tilde{z}_n - z^*\|_2 \leq R_n \leq \frac{\mu}{L_2 \xi}.$$

Therefore, the number of restarts is

$$\frac{R}{2^{n-1}} \leq \frac{\mu}{L_2 \xi} \Leftrightarrow n = \left\lceil \log \frac{L_2 R \xi}{\mu} + 1 \right\rceil.$$

Next we switch to Algorithm 3 and we need to obtain its number of iterations until convergence. Denote by ε' the accuracy of solution in terms of the merit function (20). Owing to first order Lipschitzness of $F(z)$ and the fact that $F(z^*) = 0$, we can get

$$\varepsilon' = m(z_k) = \frac{1}{2}\|F(z_k)\|_2^2 = \frac{1}{2}\|F(z_k) - F(z^*)\|_2^2 \leq \frac{L_1^2}{2}\|z_k - z^*\|_2^2. \tag{25}$$

Now we establish a connection between the solution in terms of merit function $m(z)$ and the duality gap $G(x, y)$. From (25) and (21) we get the following:

$$\varepsilon = G(x,y) = \max_{y' \in \mathbb{R}^n} f(x, y') - \min_{x' \in \mathbb{R}^n} f(x', y) \leq \frac{L_1}{\mu^2} m(z_k) = \frac{L_1}{\mu^2}\varepsilon'$$

$$\Leftrightarrow \frac{\mu^2 \varepsilon}{L_1} \leq \varepsilon'. \tag{26}$$

Then, from (23), (22), (25) and (26) we can obtain the needed number of iterations k

$$\frac{\mu^2\varepsilon}{L_1} \overset{(25),(26)}{\leq} \frac{L_1^2}{2}\|z_k - z^*\|_2^2$$

$$\overset{(23)}{\leq} \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\|z_{k-1} - z^*\|_2^2\right)^2 \leq \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\left(\frac{L_1L_2}{\mu^2}\|z_{k-2} - z^*\|_2^2\right)^2\right)^2 \leq \ldots$$

$$\leq \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\right)^{2^{k-1}-2}\|z_1 - z^*\|_2^{2^k} \overset{(22)}{\leq} \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\right)^{2^{k-1}-2}\left(\frac{\mu^2}{L_1L_2}\right)^{2^k}$$

$$\Leftrightarrow \frac{2\mu^2\varepsilon}{L_1^3} \leq \left(\frac{\mu^2}{L_1L_2}\right)^{2^{k-1}+2} \Leftrightarrow \log\frac{2\mu^2\varepsilon}{L_1^3} \leq (2^{k-1}+2)\log\frac{\mu^2}{L_1L_2}$$

Since $\log(\mu^2/L_1L_2) < 0$,

$$\log\frac{2\mu^2\varepsilon}{L_1^3} \leq 2^{k-1}\log\frac{\mu^2}{L_1L_2} \Leftrightarrow k = \left\lceil \log\frac{\log\frac{L_1^3}{2\mu^2\varepsilon}}{\log\frac{L_1L_2}{\mu^2}}\right\rceil + 1.$$

Finally, the total number of iterations of Algorithm 4 is

$$N = \sum_{i=1}^{n} T_i + k$$

$$\leq \frac{1}{2}\left(\frac{64L_pR^p}{p!\mu}\right)^{\frac{2}{p+1}}\left\lceil \log\frac{L_2\xi R}{\mu} + 1\right\rceil + \left\lceil \log\frac{\log\frac{L_1^3}{2\mu^2\varepsilon}}{\log\frac{L_1L_2}{\mu^2}}\right\rceil + 1$$

$$= O\left(\left(\frac{L_pR^p}{\mu}\right)^{\frac{2}{p+1}}\log\frac{L_2\xi R}{\mu} + \log\frac{\log\frac{L_1^3}{2\mu^2\varepsilon}}{\log\frac{L_1L_2}{\mu^2}}\right)$$

□

3.3 Gradient norm minimization

In this subsection we apply the framework from [4] to Algorithm 4, introduce Algorithm 5 for problem (3) and analyze its complexity in terms of the norm of the gradient $\|\nabla g(x, y)\|_2$.

Firstly, we need to introduce some technical lemmas.

Lemma 3.7. *If $g(x, y)$ is p-Lipchitz* (12)*, then its partial p-th order derivatives are also Lipschitz.*

$$\forall \hat{x}, x \in \mathbb{R}^n,\ \hat{y}, y \in \mathbb{R}^m \Rightarrow \|\nabla^p_{x^iy^{p-i}} g(\hat{x}, \hat{y}) - \nabla^p_{x^iy^{p-i}} g(x, y)\|_2 \leq L_p\|\hat{z} - z\|_2. \qquad (27)$$

Proof. Here we provide proof only for $\nabla^p_{x\ldots x}$. For other partial derivatives the proof is analogous.

From definition of $\|\cdot\|_2$

$$\|\nabla^p_{x\ldots x} g(\hat{x}, \hat{y}) - \nabla^p_{x\ldots x} g(x, y)\|_2 = \max_{\|s\|_2\leq 1} |(\nabla^p_{x\ldots x} g(\hat{x}, \hat{y}) - \nabla^p_{x\ldots x} g(x, y))[s]^p|$$

$$= \max_{\|s\|_2\leq 1}\left|(\nabla^p g(\hat{x}, \hat{y}) - \nabla^p g(x, y))\left[\begin{pmatrix} s \\ 0\end{pmatrix}\right]^p\right|$$

$$\leq \max_{\|h\|_2\leq 1} |(\nabla^p g(\hat{x}, \hat{y}) - \nabla^p g(x, y))[h]^p|$$

$$= \|\nabla^p g(\hat{x}, \hat{y}) - \nabla^p g(x, y)\|_2 \leq L_p\|\hat{z} - z\|_2.$$

□

Lemma 3.8. *Let* $\nabla^p_{x\ldots x} g(x,y)$ *be Lipschitz* (27)*. Then*

$$\forall n \in [p] \Rightarrow \|\nabla^{p-n}_{x\ldots x} g(\hat{z}) - \nabla^{p-n}_{x\ldots x}\Phi_{(x,y),p}(\hat{z})\|_2 \le \frac{L_p(\sqrt{2})^n}{(n+1)!}\|\hat{z}-z\|_2^{n+1}. \quad (28)$$

Proof. We prove this by induction.

The base of induction $n = 1$ follows from the definition of Taylor approximation. Denote $f(z) = \nabla^{p-1}_{x\ldots x} g(z)$.

$$\begin{aligned}
&\|\nabla^{p-1}_{x\ldots x} g(\hat{z}) - \nabla^{p-1}_{x\ldots x}\Phi_{(x,y),p}(\hat{z})\|_2 \\
&= \|\nabla^{p-1}_{x\ldots x} g(\hat{z}) - \nabla^{p-1}_{x\ldots x} g(z) - \nabla^{p}_{x\ldots xx} g(z)[\hat{x}-x] - \nabla^{p}_{x\ldots xy} g(z)[\hat{y}-y]\|_2 \\
&= \|f(\hat{z}) - f(z) - \nabla f(z)[\hat{z}-z]\|_2 \\
&= \|\int_0^1 \langle \nabla f(z+\tau(\hat{z}-z)) - \nabla f(z); \hat{z}-z\rangle d\tau\|_2 \\
&\le \int_0^1 \left\| \begin{pmatrix} \nabla^p_{x\ldots xx} g(z+\tau(\hat{z}-z)) \\ \nabla^p_{x\ldots xy} g(z+\tau(\hat{z}-z)) \end{pmatrix} - \begin{pmatrix} \nabla^p_{x\ldots xx} g(z) \\ \nabla^p_{x\ldots xy} g(z)) \end{pmatrix} \right\|_2 \|\hat{z}-z\|_2 d\tau \\
&= \int_0^1 \sqrt{\|\nabla^p_{x\ldots xx} g(z+\tau(\hat{z}-z)) - \nabla^p_{x\ldots xx} g(z)\|_2^2 + \|\nabla^p_{x\ldots xy} g(z+\tau(\hat{z}-z)) - \nabla^p_{x\ldots xy} g(z)\|_2^2} \cdot \\
&\cdot \|\hat{z}-z\|_2 d\tau \\
&\overset{(27)}{\le} \sqrt{2} L_p \|\hat{z}-z\|_2^2 \int_0^1 \tau d\tau = \frac{L_p\sqrt{2}}{2}\|\hat{z}-z\|_2^2.
\end{aligned}$$

Now assume it holds for $n = p-1$:

$$\begin{aligned}
&\|\nabla_x g(\hat{z}) - \nabla_x \Phi_{(x,y),p}(\hat{z})\|_2 \\
&= \Big\| \nabla_x g(\hat{z}) - \nabla_x g(z) - (\nabla^2_{xx} g(z)[\hat{x}-x] - \nabla^2_{xy} g(z)[\hat{y}-y]) - \ldots - \\
&-\nabla_x\left(\frac{1}{p!}\nabla^p g(z)[\hat{z}-z]^p\right)\Big\|_2 \\
&\le \frac{L_p(\sqrt{2})^{p-1}}{p!}\|\hat{z}-z\|_2^p. \quad (29)
\end{aligned}$$

And consider $n = p$

$$
\begin{aligned}
&|g(\hat{z}) - \Phi_{(x,y),p}(\hat{z})| \\
&= |g(\hat{z}) - g(z) - \nabla_x g(z)[\hat{x} - x] - \nabla_y g(z)[\hat{y} - y] - ... - \frac{1}{p!}\nabla^p g(z)[\hat{z} - z]^p| \\
&\leq \int_0^1 \left\| \begin{pmatrix} \nabla_x g(z + \tau(\hat{z} - z)) \\ \nabla_y g(z + \tau(\hat{z} - z)) \end{pmatrix} - \begin{pmatrix} \nabla_x g(z) \\ \nabla_y g(z) \end{pmatrix} - \right. \\
&-\tau \begin{pmatrix} \nabla^2_{xx} g(z)[\hat{x} - x] + \nabla^2_{xy} g(z)[\hat{y} - y] \\ \nabla^2_{yx} g(z)[\hat{x} - x] + \nabla^2_{yy} g(z)[\hat{y} - y] \end{pmatrix} - ... - \\
&\left. - \frac{\tau^{p-1}}{p!} \begin{pmatrix} \nabla_x(\nabla^p g(z)[\hat{z} - z]^p) \\ \nabla_y(\nabla^p g(z)[\hat{z} - z]^p) \end{pmatrix} \right\|_2 \|\hat{z} - z\|_2 d\tau \\
&= \int_0^1 \Big(\|\nabla_x g(z + \tau(\hat{z} - z)) - \nabla_x g(z) - \\
&-\tau(\nabla^2_{xx} g(z)[\hat{x} - x] + \nabla^2_{xy} g(z)[\hat{y} - y]) - ... - \\
&- \frac{\tau^{p-1}}{p!} \nabla_x(\nabla^p g(z)[\hat{z} - z]^p)\|_2^2 + \\
&+ \|\nabla_y g(z + \tau(\hat{z} - z)) - \nabla_y g(z) - \\
&-\tau(\nabla^2_{yx} g(z)[\hat{x} - x] + \nabla^2_{yy} g(z)[\hat{y} - y]) - ... - \\
&- \frac{\tau^{p-1}}{p!} \nabla_y(\nabla^p g(z)[\hat{z} - z]^p)\|_2^2 \Big)^{1/2} \|\hat{z} - z\|_2 d\tau.
\end{aligned}
$$

If we denote $\hat{z} = z + \tau(\hat{z} - z)$ in (29), each of two factors under the square root is indeed what we had for $n = p - 1$. Finally,

$$
\begin{aligned}
\|\nabla_x g(\hat{z}) - \nabla_x \Phi_{(x,y),p}(\hat{z})\|_2 &\leq \sqrt{2} \frac{L_p(\sqrt{2})^{p-1}}{p!} \|\hat{z} - z\|_2^{p+1} \int_0^1 \tau^p d\tau \\
&= \frac{L_p(\sqrt{2})^p}{(p+1)!} \|\hat{z} - z\|_2^{p+1}.
\end{aligned}
$$

For any other partial derivative in (28) the result is the same and can be obtained in a similar way. □

The next lemma is a modified version of Lemma 5.2 from [9] for SPP.

Lemma 3.9 (Lemma 5.2 from [9]). *Let* $(\tilde{x}, \tilde{y}) = T^g_{p,M}(x, y)$, $p \geq 2$, *where* $M \geq \sqrt{2} p L_p > \frac{1}{\sqrt{2}} p L_p$ *and assumption 4 hold. Then*

$$
\|\nabla g(\tilde{x}, \tilde{y})\|_2^{\frac{p+1}{p}} \frac{M^{\frac{3p+1}{2p}}}{2^{\frac{2p^2+p+1}{2p}} p(p+1)!} \leq g(x, \tilde{y}) - g(\tilde{x}, y). \tag{30}
$$

Proof.

$$
\|\nabla g(\tilde{x}, \tilde{y})\|_2^2 = \|\nabla_x g(\tilde{x}, \tilde{y})\|_2^2 + \|\nabla_y g(\tilde{x}, \tilde{y})\|_2^2.
$$

Firstly, consider ∇_x:

$$\begin{aligned}\|\nabla_x g(\tilde{x},\tilde{y})\|_2^2 &= \|\nabla_x g(\tilde{x},\tilde{y}) - \nabla_x \Phi_{(x,y),p}(\tilde{x},\tilde{y}) + \nabla_x \Phi_{(x,y),p}(\tilde{x},\tilde{y}) - \\ &\quad - \nabla_x \Omega_{(x,y),p,M}(\tilde{x},\tilde{y}) + \nabla_x \Omega_{(x,y),p,M}(\tilde{x},\tilde{y})\|_2^2 \\ &\leq \Big(\|\nabla_x g(\tilde{x},\tilde{y}) - \nabla_x \Phi_{(x,y),p}(\tilde{x},\tilde{y})\|_2 + \\ &\quad + \|\nabla_x \Phi_{(x,y),p}(\tilde{x},\tilde{y}) - \nabla_x \Omega_{(x,y),p,M}(\tilde{x},\tilde{y})\| + \|\nabla_x \Omega_{(x,y),p,M}(\tilde{x},\tilde{y})\|_2 \Big)^2 \\ &\leq \left(\frac{2^{\frac{p-1}{2}} L_p}{p!} \|\tilde{z} - z\|_2^p + \frac{2^{\frac{p-1}{2}} M}{p!} \|\tilde{x} - x\|_2^p \right)^2 \leq 2^p M^2 \|\tilde{z} - z\|_2^{2p}.\end{aligned}$$

For ∇_y in a similar way we get the same result

$$\|\nabla_x g(\tilde{x},\tilde{y})\|_2^2 \leq 2^p M^2 \|\tilde{z} - z\|_2^{2p}.$$

Summing these two results, we obtain

$$\|\nabla g(\tilde{x},\tilde{y})\|_2^2 \leq 2^{p+1} M \big(\|\tilde{x} - x\|_2^2 + \|\tilde{y} - y\|_2^2 \big)^p. \tag{31}$$

Secondly, consider point $(\tilde{x}, y)$. From (28) it is obvious that

$$|g(\tilde{x},y) - \Phi_{(x,y),p}(\tilde{x},y)| \leq \frac{L_p(\sqrt{2})^p}{(p+1)!} \|(\tilde{x},y) - (x,y)\|_2^{p+1} = \frac{L_p(\sqrt{2})^p}{(p+1)!} \|\tilde{x} - x\|_2^{p+1}.$$

From this fact we get

$$\begin{aligned}g(\tilde{x},y) &\leq \Phi_{(x,y),p}(\tilde{x},y) + \frac{L_p(\sqrt{2})^p}{(p+1)!} \|\tilde{x} - x\|_2^{p+1} \\ &= \Phi_{(x,y),p}(\tilde{x},y) + \frac{L_p(\sqrt{2})^{p-1}}{(p+1)!} \|\tilde{x} - x\|_2^{p+1} - \\ &\quad - \left(\frac{M(\sqrt{2})^{p-1}}{(p+1)!} \|\tilde{x} - x\|_2^{p+1} - \frac{L_p(\sqrt{2})^p}{(p+1)!} \|\tilde{x} - x\|_2^{p+1} \right) \\ &= \Omega_{(x,y),p,M}(\tilde{x},y) - (M - L_p\sqrt{2}) \frac{(\sqrt{2})^{p-1} \|\tilde{x} - x\|_2^{p+1}}{(p+1)!} \\ &\leq \Omega_{(x,y),p,M}(\tilde{x},\tilde{y}) - (M - L_p\sqrt{2}) \frac{(\sqrt{2})^{p-1} \|\tilde{x} - x\|_2^{p+1}}{(p+1)!}.\end{aligned}$$

Since $M \geq \sqrt{2} p L_p \Leftrightarrow -L_p\sqrt{2} \geq -\frac{M}{p}$. we have

$$\Omega_{(x,y),p,M}(\tilde{x},\tilde{y}) - g(\tilde{x},y) \geq \frac{M(p-1)(\sqrt{2})^{p-1} \|\tilde{x} - x\|_2^{p+1}}{p(p+1)!} \geq \frac{M \|\tilde{x} - x\|_2^{p+1}}{p(p+1)!}. \tag{32}$$

Now consider the point $(x, \tilde{y})$. In a similar way we can get the following result:

$$g(x,\tilde{y}) - \Omega_{(x,y),p,M}(\tilde{x},\tilde{y}) \geq \frac{M \|\tilde{y} - y\|_2^{p+1}}{p(p+1)!}. \tag{33}$$

From the sum of (32) and (33) we obtain

$$g(x,\tilde{y}) - g(\tilde{x},y) \geq \frac{M}{p(p+1)!}\Big(\|\tilde{x}-x\|_2^{p+1} + \|\tilde{y}-y\|_2^{p+1}\Big). \tag{34}$$

Finally, we need to connect (31) and (34). From Hölder's inequality we can get

$$\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}} \leq n^{\frac{q-p}{qp}} \left(\sum_{i=1}^n x_i^q\right)^{\frac{1}{q}},$$

where $q, p \in \mathbb{N},\ q > p \geq 1$. Now, from (31) it follows that

$$\left(\frac{\|\nabla g(\tilde{x},\tilde{y})\|_2^2}{2^{p+1}M}\right)^{\frac{1}{2p}} \leq \left(\|\tilde{x}-x\|_2^2 + \|\tilde{y}-y\|_2^2\right)^{\frac{1}{2}}.$$

And, from (34) we can get

$$\left(\frac{p(p+1)!(g(x,\tilde{y}) - g(\tilde{x},y))}{M}\right)^{\frac{1}{p+1}} \geq \left(\|\tilde{x}-x\|_2^{p+1} + \|\tilde{y}-y\|_2^{p+1}\right)^{\frac{1}{p+1}}.$$

Since $p \geq 2$, we obtain the final result

$$\|\nabla g(\tilde{x},\tilde{y})\|_2^{\frac{p+1}{p}} \frac{M^{\frac{3p+1}{2p}}}{2^{\frac{2p^2+p+1}{2p}} p(p+1)!} \leq g(x,\tilde{y}) - g(\tilde{x},y).$$

□

Now we have all the needed information to estimate the final convergence rate of the Algorithm 5 for gradient norm minimization.

Theorem 3.10. *Assume the function $g(x,y) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is convex by x and concave by y, p times differentiable on $\mathbb{R}^n$ with L_p-Lipschitz p-th derivative. Let $\tilde{z}$ be generated by Algorithm 5. Then*

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

and the total complexity of Algorithm 5 is

$$O\left(\left(\frac{L_p R^p}{\varepsilon}\right)^{\frac{2}{p+1}} \log \frac{L_2 R^2 \xi}{\varepsilon}\right),$$

where $\xi = \max\left\{1, \frac{4RL_1}{\varepsilon}\right\}$.

Proof. Denote $z_\mu^* = (x_\mu^*, y_\mu^*)$ the saddle point of $g_\mu(z)$. First of all, since $g_\mu(x,y)$ is strongly-convex-strongly-concave function, we can apply restart technique to it every time the distance to its saddle point $\|z - z_\mu^*\|_2$ reduces twice. To check this, we consider upper estimate of the distance to the solution of regular function $R : R \geq \|z^* - z\|_2$ and show, that on each i-th restart $\|z_\mu^* - z_i\|_2 \leq \|z^* - z_i\|_2 \leq R_i$. We prove this by induction.

$$\begin{aligned} g(x_\mu^*, y_1) + \frac{\mu}{2}\|x_\mu^* - x_1\|_2^2 = g_\mu(x_\mu^*, y_1) \leq g_\mu(x^*, y_1) &= g(x^*, y_1) + \frac{\mu}{2}\|x^* - x_1\|_2^2 \\ &\leq g(x_\mu^*, y_1) + \frac{\mu}{2}\|x^* - x_1\|_2^2 \end{aligned}$$

Algorithm 5 Restarted HighOrderMirrorProx with local quadratic convergence for gradient norm minimization

1: **Input** $z_1 \in \mathcal{Z}, p \geq 1, 0 < \varepsilon < 1, R : R \geq \|z_1 - z^*\|_2$, $\rho \in (0,1)$, $\alpha \in (0,1)$.
2: **Define**:

$$\tilde{z}_1 = z_1, \quad M = \sqrt{2}pL_p, \quad \mu = \frac{\varepsilon}{4R}, \quad \xi = \max\left\{1, \frac{4RL_1}{\varepsilon}\right\},$$
$$\varepsilon' = \frac{M^{\frac{3p+1}{2p}}\varepsilon^{\frac{p+1}{p}}}{2^{\frac{2p^2+3p+3}{2p}}p(p+1)!},$$
$$g_\mu(x,y) = g(x,y) + \frac{\mu}{2}\left(\|x - x_1\|_2^2 - \|y - y_1\|_2^2\right).$$

3: **for** $i \in [n]$, where $n = \left\lceil \log \frac{L_2 R\xi}{\mu} + 1 \right\rceil$ **do**
4: Set $R_i = \frac{R}{2^{i-1}}$
5: Set $T_i = \left\lceil \left(\frac{64L_p R_i^{p-1}}{p!\mu}\right)^{\frac{2}{p+1}} \right\rceil$
6: Run Algorithm 1 for g_μ with $\tilde{z}_i$, p, T_i as input
7: $\tilde{z}_{i+1} = \bar{z}_{T_i}$
8: **end for**
9: Run Algorithm 3 with $\tilde{z}_{i+1}$, ε', $\bar{\gamma} = \frac{L_2\mu^2}{2L_1^2}$, ρ, α, g_μ as input
10: **Find** $\tilde{z} = T_{p,\,M}^{g_\mu}(z_k)$
11: **Output** $\tilde{z}$.

$$\Leftrightarrow \|x_\mu^* - x_1\|_2 \leq \|x^* - x_1\|_2.$$

$$g(x_1, y_\mu^*) - \frac{\mu}{2}\|y_\mu^* - y_1\|_2 = g_\mu(x_1, y_\mu^*) \geq g_\mu(x_1, y^*) = g(x_1, y^*) - \frac{\mu}{2}\|y^* - y_1\|_2^2$$
$$\geq g(x_1, y_\mu^*) - \frac{\mu}{2}\|y^* - y_1\|_2^2$$

$$\Leftrightarrow \|y_\mu^* - y_1\|_2 \leq \|y^* - y_1\|_2.$$

This gives us

$$\|z_\mu^* - z_1\|_2 \leq \|z^* - z_1\|_2 \leq R.$$

Now suppose, that $\|z_\mu^* - z_i\|_2 \leq \|z^* - z_i\|_2 \leq R_i = R/2^{i-1}$. Consider $i+1$. From the proof of Theorem 3.3 and our choice of T_i in Algorithm 5, we know, that

$$\mu\|z_{i+1} - z_\mu^*\|_2^2 = \mu\|\bar{z}_{T_i} - z_\mu^*\|_2^2 \leq \frac{16L_p}{p!}\left(\frac{R_i^2}{2T_i}\right)^{\frac{p+1}{2}} \leq \mu R_{i+1}^2$$
$$\Leftrightarrow \|z_{i+1} - z_\mu^*\|_2 \leq R_{i+1}.$$

Then, we need to find the number of restarts n of Algorithm 2 to reach the area of local quadratic convergence of Algorithm 3: $\|\tilde{z}_n - z^*\|_2 \leq \frac{\mu}{L_2\xi}$. We can choose such n, that

$$\|\tilde{z}_n - z^*\|_2 \leq R_n \leq \frac{\mu}{L_2\xi},$$

where $\xi = \max\left\{1, \frac{L_1}{\mu}\right\} = \max\left\{1, \frac{4RL_1}{\varepsilon}\right\}$. Therefore, the number of restarts is

$$\frac{R}{2^{n-1}} \le \frac{\mu}{L_2\xi} \Leftrightarrow n = \left\lceil \log\frac{L_2R\xi}{\mu} + 1 \right\rceil.$$

Next, we need to show, that Algorithm 5 converges in terms of $\|\nabla g_\mu(z)\|_2$. Let $\tilde z = (\tilde x, \tilde y)$ be the output of Algorithm 5. From the definition of g_μ we get

$$\begin{aligned}
\|\nabla g(\tilde x,\tilde y)\|_2^2 &= \|\nabla_x g_\mu(\tilde x,\tilde y) - \mu(\tilde x - x_1)\|_2^2 + \|\nabla_y g_\mu(\tilde x,\tilde y) + \mu(\tilde y - y_1)\|_2^2 \\
&\le \left(\|\nabla_x g_\mu(\tilde x,\tilde y)\|_2 + \mu\|\tilde x - x\|_2\right)^2 + \left(\|\nabla_y g_\mu(\tilde x,\tilde y)\|_2 + \mu\|\tilde y - y\|_2\right)^2 \\
&\le 2\left(\|\nabla_x g_\mu(\tilde x,\tilde y)\|_2^2 + \|\nabla_y g_\mu(\tilde x,\tilde y)\|_2^2\right) + 2\mu^2\left(\|\tilde x - x\|_2^2 + \|\tilde y - y\|_2^2\right) \\
&= 2\|\nabla g_\mu(\tilde x,\tilde y)\|_2^2 + 2\mu^2\|\tilde z - z_1\|_2^2
\end{aligned}$$

$$\Leftrightarrow \|\nabla g(\tilde x,\tilde y)\|_2 \le \sqrt{2\|\nabla g_\mu(\tilde x,\tilde y)\|_2^2 + 2\mu^2\|\tilde z - z_1\|_2^2}.$$

Firstly, we estimate $\|\nabla g_\mu(\tilde x,\tilde y)\|_2$. From (30) we know, that

$$\begin{aligned}
\|\nabla g_\mu(\tilde x,\tilde y)\|_2^{\frac{p+1}{p}} \frac{M^{\frac{3p+1}{2p}}}{2^{\frac{2p^2+p+1}{2p}}p(p+1)!} &\overset{(30)}{\le} g_\mu(x,\tilde y) - g_\mu(\tilde x, y) \\
&\le \max_{\tilde y\in\mathbb{R}^m} g_\mu(x,\tilde y) - \min_{\tilde x\in\mathbb{R}^n} g_\mu(\tilde x, y) = G_\mu(x,y) \le \varepsilon'.
\end{aligned}$$

$$\Leftrightarrow \|\nabla g_\mu(\tilde x,\tilde y)\|_2 \le \left(\frac{2^{\frac{2p^2+p+1}{2p}}p(p+1)!\varepsilon'}{M^{\frac{3p+1}{2p}}}\right)^{\frac{p}{p+1}} = \frac{\varepsilon}{2}. \tag{35}$$

Secondly, we estimate $\mu\|\tilde z - z_1\|_2$. By definition of R we know, that

$$\|z^* - z_1\|_2 \le R.$$

And since $\tilde z$ is closer to solution then z_1, we have

$$\|\tilde z - z^*\|_2 \le \|z^* - z_1\|2 \le R.$$

From these facts and triangle inequality we get

$$\mu\|\tilde z - z_1\|_2 \le \mu\left(\|\tilde z - z^*\|_2 + \|z^* - z_1\|_2\right) \le 2R\mu = \frac{\varepsilon}{2}. \tag{36}$$

Thus, from (35) and (36) we obtain

$$\|\nabla g_\mu(\tilde x,\tilde y)\|_2 \le \sqrt{2\varepsilon^2/4 + 2\varepsilon^2/4} = \varepsilon.$$

Finally, we need to estimate complexity of the Algorithm 5.

$$\begin{aligned}
N = \sum_{i=1}^n T_i + k &\le \left(\frac{64L_p}{p!\mu}\right)^{\frac{2}{p+1}} \sum_{i=1}^n R_i^{\frac{2(p-1)}{p+1}} + n + k \\
&\le \left(\frac{64L_pR^{p-1}}{p!\mu}\right)^{\frac{2}{p+1}} \cdot n + n + k \\
&= O\left(\left(\frac{L_pR^p}{\varepsilon}\right)^{\frac{2}{p+1}} \log\frac{L_2R^2\xi}{\varepsilon}\right),
\end{aligned}$$

where $\xi = \max\left\{1, \frac{4RL_1}{\varepsilon}\right\}$. Here k is the number of iterations of Algorithm 3 inside Algorithm 5. We dropped it due to its $\log\log$ dependence on ε.

□

4 Discussion

In this work we propose three methods for p-th order tensor methods for strongly-convex-strongly-concave SPP. Two of these methods tackle classical minimax SPP (1) and MVI (2) problems, and the third method aims at gradient norm minimization of SPP (3).

The methods for minimax problem are based on the ideas, developed in the works [2] and [11]. In [2] the authors use p-th order oracle to construct an algorithm for MVI problems with monotone operator. As a corollary, this algorithm allows to solve SPP with convex-concave objective. Because of strong convexity and strong concavity of our problem, we can apply a restart technique to the method from [2] and get better algorithm complexity. To further improve local convergence rate we switch to the algorithm from [11] in the area of its quadratic convergence. This way we get rid of the multiplicative logarithmic factor and get additive $\log\log$ factor in the final complexity estimate and get locally quadratic convergence.

The method for gradient norm minimization relies on the works [9] and [4]. From [9] we take the result, that connects norm of the gradient of the objective with objective residual, and slightly modify it for SPP. This step allows us to use the framework from [4] and use our optimal algorithm for minimax SPP for gradient norm minimization.

In spite of all the improvements, we should remind about many additional assumptions about the problem, which reduces number of real problems, that can suit to it.

One of possible directions for further research are the more general Hölder conditios instead of Lipschitz conditions and uniformly convex case. Additionally, the author in [2] provided implementation details of the Algorithm 1 only for $p = 2$. Therefore, the questions about its realizaition for $p > 2$ are still opened.

References

[1] Brian Bullins. Fast minimization of structured convex quartics. *arXiv preprint arXiv:1812.10349*, 2018.

[2] Brian Bullins and Kevin A Lai. Higher-order methods for convex-concave min-max optimization and monotone variational inequalities. *arXiv preprint arXiv:2007.04528*, 2020.

[3] Brian Bullins and Richard Peng. Higher-order accelerated methods for faster non-smooth optimization. *arXiv preprint arXiv:1906.01621*, 2019.

[4] Pavel Dvurechensky, Alexander Gasnikov, Petr Ostroukhov, César A Uribe, and Anastasiya Ivanova. Near-optimal tensor methods for minimizing the gradient norm of convex function. *arXiv preprint arXiv:1912.03381*, 2019.

[5] Alexander Gasnikov, Darina Dvinskikh, Pavel Dvurechensky, Dmitry Kamzolov, Dmitry Pasechnyk, Vladislav Matykhin, Nazarii Tupitsa, and Alexei Chernov. Accelerated meta-algorithm for convex optimization. *Computational Mathematics and Mathematical Physics*, 61(1), 2020.

[6] Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, and César A. Uribe. Optimal tensor methods in smooth convex and uniformly convex optimization. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1374–1391, Phoenix, USA, 25–28 Jun 2019. PMLR. arXiv:1809.00382.

[7] Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, César A. Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Near optimal methods for minimizing convex functions with lipschitz p-th derivatives. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1392–1393, Phoenix, USA, 25–28 Jun 2019. PMLR.

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

[9] Geovani Nunes Grapiglia and Yurii Nesterov. Tensor methods for finding approximate stationary points of convex functions. *arXiv preprint arXiv:1907.07053*, 2019.

[10] K. H. Hoffmann and H. J. Kornstaedt. Higher-order necessary conditions in abstract mathematical programming. *Journal of Optimization Theory and Applications*, 26(4):533–568, Dec 1978.

[11] Kevin Huang, Junyu Zhang, and Shuzhong Zhang. Cubic regularized newton method for saddle point models: a global and local convergence analysis. *arXiv preprint arXiv:2008.09919*, 2020.

[12] Guy Kornowski and Ohad Shamir. High-order oracle complexity of smooth and strongly convex optimization. *arXiv preprint arXiv:2010.06642*, 2020.

[13] Galina Korpelevich. The extragradient method for finding saddle points and other problems. *Eknomika i Matematicheskie Metody*, 12:747–756, 1976.

[14] Tianyi Lin, Chi Jin, Michael Jordan, et al. Near-optimal algorithms for minimax optimization. *arXiv preprint arXiv:2002.02417*, 2020.

[15] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020.

[16] Renato DC Monteiro and Benar F Svaiter. Iteration-complexity of a newton proximal extragradient method for monotone variational inequalities and inclusion problems. *SIAM Journal on Optimization*, 22(3):914–935, 2012.

[17] Renato DC Monteiro and Benar Fux Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6):2755–2787, 2010.

[18] Arkadi Nemirovski. 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, 2004.

[19] A.S. Nemirovsky and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. J. Wiley & Sons, New York, 1983.

[20] Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3):319–344, 2007. First appeared in 2003 as CORE discussion paper 2003/68.

[21] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[22] Yurii Nesterov and Boris Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[23] Yurii Nesterov and Laura Scrimali. Solving strongly monotone variational and quasi-variational inequalities. *Available at SSRN 970903*, 2006.

[24] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

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

[26] Fedor Stonyakin, Alexander Gasnikov, Pavel Dvurechensky, Mohammad Alkousa, and Alexander Titov. Generalized Mirror Prox for monotone variational inequalities: Universality and inexact oracle. *arXiv:1806.05140*, 2018.

[27] Paul Tseng. On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 60(1-2):237–252, 1995.

[28] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, MIT, 2008.

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