КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2022 Т. 14 № 2 С. 357–376
DOI: 10.20537/2076-7633-2022-14-2-357-376

Ки&М

МАТЕМАТИЧЕСКИЕ ОСНОВЫ И ЧИСЛЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ

УДК: 519.853.62

Тензорные методы для сильно выпуклых сильно вогнутых седловых задач и сильно монотонных вариационных неравенств

П. А. Остроухов[1,2,a], Р. А. Камалов[1,3,b], П. Е. Двуреченский[4,c], А. В. Гасников[1,2,5,d]

[1]Московский физико-технический институт,
Россия, 141701, Московская область, г. Долгопрудный, Институтский переулок, д. 9
[2]Институт проблем передачи информации им. А. А. Харкевича Российской академии наук,
Россия, 127051, г. Москва, Большой Каретный переулок, д. 19, стр. 1
[3]Институт проблем управления им. В. А. Трапезникова Российской академии наук,
Россия, 117997, г. Москва, ул. Профсоюзная, д. 65
[4]Институт прикладного анализа и стохастики им. Вейерштрасса,
Германия, 10117, г. Берлин, Моренштрассе, д. 39
[5]Кавказский математический центр, Адыгейский государственный университет,
Россия, 385000, Республика Адыгея, г. Майкоп, ул. Первомайская, д. 208

E-mail: [a] ostroukhov@phystech.edu, [b] kamalov.ra@phystech.edu, [c] pavel.dvurechensky@wias-berlin.de,
[d] gasnikov@yandex.ru

Получено 12.02.2022.
Принято к публикации 13.02.2022.

В данной статье предлагаются методы оптимизации высокого порядка (тензорные методы) для решения двух типов седловых задач. Первый тип — это классическая мин-макс-постановка для поиска седловой точки функционала. Второй тип — это поиск стационарной точки функционала седловой задачи путем минимизации нормы градиента этого функционала. Очевидно, что стационарная точка не всегда совпадает с точкой оптимума функции. Однако необходимость в решении подобного типа задач может возникать в случае, если присутствуют линейные ограничения. В данном случае из решения задачи поиска стационарной точки двойственного функционала можно восстановить решение задачи поиска оптимума прямого функционала. В обоих типах задач какие-либо ограничения на область определения целевого функционала отсутствуют. Также мы предполагаем, что целевой функционал является μ-сильно выпуклым и μ-сильно вогнутым, а также что выполняется условие Липшица для его p-й производной.

Для задач типа «мин-макс» мы предлагаем два алгоритма. Так как мы рассматриваем сильно выпуклую и сильно вогнутую задачу, первый алгоритм использует существующий тензорный метод для решения выпуклых вогнутых седловых задач и ускоряет его с помощью техники рестартов. Таким образом удается добиться линейной скорости сходимости. Используя дополнительные предположения о выполнении условий Липшица для первой и второй производных целевого функционала, можно дополнительно ускорить полученный метод. Для этого можно «переключиться» на другой существующий метод для решения подобных задач в зоне его квадратичной локальной сходимости. Так мы получаем второй алгоритм, обладающий глобальной линейной сходимостью и локальной квадратичной сходимостью.

Наконец, для решения задач второго типа существует определенная методология для тензорных методов в выпуклой оптимизации. Суть ее заключается в применении специальной «обертки» вокруг оптимального метода высокого порядка. Причем для этого условие сильной выпуклости не является необходимым. Достаточно лишь правильным образом регуляризовать целевой функционал, сделав его таким образом сильно выпуклым и сильно вогнутым. В нашей работе мы переносим эту методологию на выпукло-вогнутые функционалы и используем данную «обертку» на предлагаемом выше алгоритме с глобальной линейной сходимостью и локальной квадратичной сходимостью.

Так как седловая задача является частным случаем монотонного вариационного неравенства, предлагаемые методы также подойдут для поиска решения сильно монотонных вариационных неравенств.

Ключевые слова: вариационное неравенство, седловая задача, гладкость высокого порядка, тензорные методы, минимизация нормы градиента

Работа П. Остроухова и А. Гасникова выполнена при поддержке Министерства науки и высшего образования Российской Федерации (госзадание), № 075-00337-20-03, номер проекта 0714-2020-0005.

COMPUTER RESEARCH AND MODELING
2022 VOL. 14 NO. 2 P. 357–376
DOI: 10.20537/2076-7633-2022-14-2-357-376

Ки&М

MATHEMATICAL MODELING AND NUMERICAL SIMULATION

UDC: 519.853.62

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

P. A. Ostroukhov[1,2,a], R. A. Kamalov[1,3,b], P. E. Dvurechensky[4,c], A. V. Gasnikov[1,2,5,d]

[1]Moscow Institute of Physics and Technology,
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia
[2]Institute for Information Transmission Problems of Russian Academy of Sciences,
19/1 Bolshoy Karetny per., Moscow, 127051, Russia
[3]V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences,
65 Profsoyuznaya st., Moscow, 117997, Russia
[4]Weierstrass Institute for Applied Analysis and Stochastics,
39 Mohrenstraße, Berlin, 10117, Germany
[5]Caucasus Mathematical Center, Adyghe State University,
208 Pervomayskaya st., Maykop, Adyghe, 385000, Russia

E-mail: [a] ostroukhov@phystech.edu, [b] kamalov.ra@phystech.edu, [c] pavel.dvurechensky@wias-berlin.de,
[d] gasnikov@yandex.ru

Received 12.02.2022.
Accepted for publication 13.02.2022.

In this paper we propose high-order (tensor) methods for two types of saddle point problems. Firstly, we consider the classic min-max saddle point problem. Secondly, we consider the search for a stationary point of the saddle point problem objective by its gradient norm minimization. Obviously, the stationary point does not always coincide with the optimal point. However, if we have a linear optimization problem with linear constraints, the algorithm for gradient norm minimization becomes useful. In this case we can reconstruct the solution of the optimization problem of a primal function from the solution of gradient norm minimization of dual function. In this paper we consider both types of problems with no constraints. Additionally, we assume that the objective function is μ-strongly convex by the first argument, μ-strongly concave by the second argument, and that the p-th derivative of the objective is Lipschitz-continous.

For min-max problems we propose two algorithms. Since we consider strongly convex a strongly concave problem, the first algorithm uses the existing tensor method for regular convex concave saddle point problems and accelerates it with the restarts technique. The complexity of such an algorithm is linear. If we additionally assume that our objective is first and second order Lipschitz, we can improve its performance even more. To do this, we can switch to another existing algorithm in its area of quadratic convergence. Thus, we get the second algorithm, which has a global linear convergence rate and a local quadratic convergence rate.

Finally, in convex optimization there exists a special methodology to solve gradient norm minimization problems by tensor methods. Its main idea is to use existing (near-)optimal algorithms inside a special framework. I want to emphasize that inside this framework we do not necessarily need the assumptions of strong convexity, because we can regularize the convex objective in a special way to make it strongly convex. In our article we transfer this framework on convex-concave objective functions and use it with our aforementioned algorithm with a global linear convergence and a local quadratic convergence rate.

Since the saddle point problem is a particular case of the monotone variation inequality problem, the proposed methods will also work in solving strongly monotone variational inequality problems.

Keywords: variational inequality, saddle point problem, high-order smoothness, tensor methods, gradient norm minimization

Citation: *Computer Research and Modeling*, 2022, vol. 14, no. 2, pp. 357–376.

The research of P. Ostroukhov and A. Gasnikov is supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye), No. 075-00337-20-03, project No. 0714-2020-0005.

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 $g\colon \mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ is 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

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

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 the 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 an unconstrained case with $\mathcal{X}=\mathbb{R}^n$ and $\mathcal{Y}=\mathbb{R}^m$. Additionally, we assume that $g(x, y)$ 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 a convex-concave setting [Korpelevich, 1976; Tseng, 1995; Nemirovski, 2004; Nesterov, 2007; Tseng, 2008]. One of the most popular among first-order methods for this setting is the Mirror-Prox algorithm [Nemirovski, 2004], which treats saddle-point problems via solving the corresponding MVI. According to [Nemirovsky, Yudin, 1983], this method achieves optimal complexity of $O\left(\frac{1}{\varepsilon}\right)$ iterations for first-order methods applied to smooth convex-concave SPP in large dimensions.

The additional condition of strong convexity and strong concavity leads to better results. The algorithms from [Rockafellar, 1976; Tseng, 1995; Nesterov, Scrimali, 2006; Gidel et al., 2018; Mokhtari, Ozdaglar, Pattathil, 2020] achieve iteration complexity of $O\left(L/\mu\log\left(\frac{1}{\varepsilon}\right)\right)$. In [Lin et al., 2020] the authors proposed an algorithm with complexity $O\left(L/\sqrt{\mu_x\mu_y}\log^3\left(\frac{1}{\varepsilon}\right)\right)$, which matches up to a logarithmic factor the lower bound, obtained in [Zhang, Hong, Zhang, 2019]. It is worth mentioning that $\log^3\left(\frac{1}{\varepsilon}\right)$ factor can be improved, namely, it is possible to achieve iteration complexity of $O\left(L/\sqrt{\mu_x\mu_y}\log\left(\frac{1}{\varepsilon}\right)\right)$ (see [Gasnikov et al., 2020]).

The methods listed above use first-order oracles, and it is known from optimization that tensor methods, which use higher-order derivatives, have a faster convergence rate, yet at the cost of more expensive iteration. The idea of using derivatives of high order in optimization is not new (see [Hoffmann, Kornstaedt, 1978]). The most common type of high-order methods uses second-order oracles, for example, Newton method [Nocedal, Wright, 2006; Nesterov, Nemirovskii, 1994] and its modifications such as the cubic regularized Newton method [Nesterov, Polyak, 2006]. Recently the idea of exploiting oracles beyond the second order started to attract increased attention, especially in convex optimization [Bullins, 2018; Bullins, Peng, 2019; Gasnikov et al., 2019a; Gasnikov et al., 2019b; Dvurechensky et al., 2019].

However, much less is known about high-order methods for SPP and MVIs. In [Monteiro, Svaiter, 2012] the authors propose a second-order method based on their Hybrid Proximal Extragradient framework [Monteiro, Svaiter, 2010]. The resulting complexity is $O\left(\frac{1}{\varepsilon^{2/3}}\right)$. A recent work [Bullins, Lai, 2020] shows how to modify the Mirror-Prox method using oracles beyond second order and improves complexity to reach duality gap ε to $O\left(\frac{1}{\varepsilon^{2/(p+1)}}\right)$ for convex-concave problems with p-th order Lipschitz derivatives. The paper [Huang, Zhang, Zhang, 2020] proposes a cubic regularized Newton method for

solving SPP, which has a global linear and a 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 the next step and propose a tensor method for strongly monotone variational inequalities and, as a corollary, a tensor method for saddle point problems with a strongly-convex-strongly-concave objective. Based on the ideas from [Bullins, Lai, 2020] and [Huang, Zhang, Zhang, 2020], our work can be split into three parts.

Firstly, we apply the restart technique [Stonyakin et al., 2018] to the HighOrderMirrorProx Algorithm 1 from [Bullins, Lai, 2020], which is possible because of the strong convexity and strong concavity of the objective. Such a modification improves the algorithm complexity to $O\left(\left(\frac{L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \log \frac{\mu R^2}{\varepsilon_G}\right)$, 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, and ε_G is the error in terms of duality gap.

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 [Huang, Zhang, Zhang, 2020] to obtain the local superlinear convergence of our algorithm. The total complexity of the final Algorithm 4 becomes

$$O\left(\left(\frac{L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \log \frac{L_2 R \max\left\{1, \frac{L_1}{\mu}\right\}}{\mu} + \log \frac{\log \frac{L_1^3}{2\mu^2 \varepsilon_G}}{\log \frac{L_1 L_2}{\mu^2}}\right),$$

where L_1 and L_2 are Lipschitz constants for first and second-order derivatives, respectively. We want to emphasize that the obtained the $\log \log \left(\frac{1}{\varepsilon}\right)$ dependency on ε cannot be improved even in convex optimization [Kornowski, Shamir, 2020].

Thirdly, we apply the framework from [Dvurechensky et al., 2019] to Algorithm 4 to solve the problem (3) and obtain Algorithm 5. Its convergence rate is $\widetilde{O}\left(\left(\frac{L_p R^p}{\varepsilon_\nabla}\right)^{2/(p+1)}\right)$, where by tilde we mean an additional multiplicative log factor, and by ε_∇ an error in terms of the gradient norm of the objective.

Our paper is organized as follows. First of all, in «Preliminaries» we provide necessary notations and conditions. Then, we present the new algorithm and obtain its convergence rate in «Main Results». Firstly, in the subsection «Restarted HighOrderMirrorProx» we talk only about the restarted algorithm from [Bullins, Lai, 2020] and get its complexity. Secondly, in the subsection «Local quadratic convergence» we describe how to connect it to Algorithm 3 from [Huang, Zhang, Zhang, 2020] in its quadratic convergence area and get the final Algorithm 4 convergence rate. Thirdly, in the subsection «Gradient norm minimization» we focus on how to wrap Algorithm 4 in a framework from [Dvurechensky et al., 2019] and obtain its complexity. Finally, in «Discussion» we discuss our results and present some possible directions for future work.

Preliminaries

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

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

or equivalently

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

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

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

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

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

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

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

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

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

Denote its min-max point by

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

As we mentioned earlier, in this paper we consider two types of SPP: the classical minimax problem (1) and the gradient norm minimization (3). We need to introduce the definitions of approximate solutions of these problems. We use different indices in error notations for these problems to avoid ambiguity.

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

$$G_{\mathcal{X} \times \mathcal{Y}}(x, y) := \max_{y' \in \mathcal{Y}} g(x, y') - \min_{x' \in \mathcal{X}} g(x', y). \tag{5}$$

Since in our case $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \mathbb{R}^m$, we drop the notations of these sets from the index of the duality gap and denote the duality gap just as $G(x, y)$. Then we define the ε_G-approximate solution of (1):

$$\widetilde{x}^* \in \mathbb{R}^n, \ \widetilde{y}^* \in \mathbb{R}^m \Rightarrow G(\widetilde{x}^*, \widetilde{y}^*) \leqslant \varepsilon_G. \tag{6}$$

Secondly, for the problem (3) we don't need any additional functionals, and the ε_∇-approximate solution of (3) of the form

$$\widetilde{x}^* \in \mathbb{R}^n, \ \widetilde{y}^* \in \mathbb{R}^m \Rightarrow \|\nabla g(\widetilde{x}^*, \widetilde{y}^*)\|_2 \leqslant \varepsilon_\nabla. \tag{7}$$

Conditions

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

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

$$\forall x_1,\, x_2 \in \mathbb{R}^n,\ y \in \mathbb{R}^m \Rightarrow \langle \nabla_x g(x_1, y) - \nabla_x g(x_2, y),\, x_1 - x_2 \rangle \geqslant \mu \|x_1 - x_2\|_2^2, \tag{8}$$

$$\forall y_1,\, y_2 \in \mathbb{R}^m,\ x \in \mathbb{R}^n \Rightarrow \langle -\nabla_y g(x, y_1) + \nabla_y g(x, y_2),\, y_1 - y_2 \rangle \geqslant \mu \|y_1 - y_2\|_2^2. \tag{9}$$

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

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

$$\langle F(z_1) - F(z_2),\, z_1 - z_2 \rangle \geqslant \mu \|z_1 - z_2\|_2^2. \tag{10}$$

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

$$F(z) = F(x, y) := \begin{pmatrix} \nabla_x g(x, y) \\ -\nabla_y g(x, y) \end{pmatrix}. \tag{11}$$

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 (11). In our work we use the following conditions.

Condition 2. *$F(z)$ satisfies the first-order Lipschitz condition*:

$$\|F(z_1) - F(z_2)\|_2 \leqslant L_1 \|z_1 - z_2\|_2 \Leftrightarrow \|\nabla g(z_1) - \nabla g(z_2)\|_2 \leqslant L_1 \|z_1 - z_2\|_2. \tag{12}$$

Condition 3. *$F(z)$ satisfies the second-order Lipschitz condition*:

$$\|\nabla F(z_1) - \nabla F(z_2)\|_2 \leqslant L_2 \|z_1 - z_2\|_2 \Leftrightarrow \|\nabla^2 g(z_1) - \nabla^2 g(z_2)\|_2 \leqslant L_2 \|z_1 - z_2\|_2. \tag{13}$$

Condition 4. *$F(z)$ satisfies the pth-order Lipschitz condition* (*p-smooth*):

$$\|\nabla^{p-1} F(z_1) - \nabla^{p-1} F(z_2)\|_2 \leqslant L_p \|z_1 - z_2\|_2 \Leftrightarrow \|\nabla^p g(z_1) - \nabla^p g(z_2)\|_2 \leqslant L_p \|z_1 - z_2\|_2. \tag{14}$$

It should be noted that, to be consistent with [Bullins, Lai, 2020], we define the pth-order smoothness (Lipschitzness) of F as a property of the $(p-1)$th derivative of F, and, therefore, as a property of pth derivative of g.

Main results

Firstly, in this paragraph we propose an algorithm for finding an ε_G-approximate solution to problem (6), where $g\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is p-smooth and μ-strongly-convex-concave (conditions 4 and 1), which allows us to achieve the iteration complexity of $O\left(\left(\frac{L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \log \frac{\mu R^2}{\varepsilon_G}\right)$, where $R \geqslant \|z_1 - z^*\|_2$. This algorithm is a restarted modification of Algorithm 1.

Secondly, we develop an algorithm for tackling the same problem, where g is the first, second and pth-order Lipschitz and μ-strongly-convex-concave function (all conditions 1, 2, 3, 4). It involves the idea of exploiting the previous algorithm and then switching to Algorithm 3 in its quadratic convergence area. Thus, we obtain Algorithm 4, which allows us to achieve the iteration complexity of

$$O\left(\left(\frac{L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \log \frac{L_2 R \max\left\{1, \frac{L_1}{\mu}\right\}}{\mu} + \log \frac{\log \frac{L_1^3}{2\mu^2 \varepsilon_G}}{\log \frac{L_1 L_2}{\mu^2}}\right).$$

Thirdly, we propose an algorithm to find an ε_∇-approximate solution to problem (7), where all the conditions 1, 2, 3, 4 hold. To achieve this we use Algorithm 4, which we mentioned previously, within the framework of [Dvurechensky et al., 2019]. The final complexity of such an algorithm in terms of the norm of the gradient is $\widetilde{O}\left(\left(\frac{L_p R^p}{\varepsilon_\nabla}\right)^{2/(p+1)}\right)$, where by tilde we mean the additional multiplicative log factor.

Restarted HighOrderMirrorProx

As mentioned above, in this subparagraph we provide the restarted modification of Algorithm 1. But initially we need to give some additional information from [Bullins, Lai, 2020].

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

$$z^* \in \mathcal{Z}\colon \ \forall z \in \mathcal{Z} \Rightarrow \langle F(z), z^* - z\rangle \leqslant \varepsilon. \tag{15}$$

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

$$\forall z \in \mathcal{Z} \Rightarrow \frac{1}{\Gamma_T} \sum_{t=1}^{T} \gamma_t \langle F(z_t), z_t - z\rangle \leqslant \varepsilon, \tag{16}$$

where points z_t and $\gamma_t > 0$ are produced by Algorithm 1, and $\Gamma_T = \sum\limits_{t=1}^{T} \gamma_t$. The following lemma establishes the relation between (15) and (16).

Lemma 1 (Lemma 2.7 from [Bullins, Lai, 2020]). *Let $F\colon \mathcal{Z} \to \mathbb{R}^n$, be monotone, $z_t \in \mathcal{Z}$, $t = 1, \ldots, T$, and let $\gamma_t > 0$. Let $\overline{z}_t = \frac{1}{\Gamma_T} \sum\limits_{t=1}^{T} \gamma_t z_t$. Assume that* (16) *holds. Then $\overline{z}_t$ is an ε-approximate solution to* (2).

The MVI problem (2), which is sometimes called «weak MVI», is closely connected to the strong MVI problem, where we need to find

$$z^* \in \mathcal{Z}\colon \ \forall z \in \mathcal{Z} \Rightarrow \langle F(z^*), z^* - z\rangle \leqslant 0. \tag{17}$$

If F is continuous and monotone, the problems (2) and (17) are eqiuvalent.

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

Algorithm 1. HighOrderMirrorProx [Algorithm 1 in [Bullins, Lai, 2020]]

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

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

  Define $\Gamma_T \overset{def}{=} \sum\limits_{t=1}^{T} \gamma_t$
return $\overline{z}_T \overset{def}{=} \frac{1}{\Gamma_T} \sum\limits_{t=1}^{T} \gamma_t \widehat{z}_t$.

Lemma 2 (Lemma 4.1 from [Bullins, Lai, 2020]). *Suppose $F\colon \mathcal{Z} \to \mathbb{R}^n$ is pth-order Lipschitz and let $\Gamma_T = \sum\limits_{t=1}^{T} \gamma_t$. Then the iterates $\{\widehat{z}_t\}_{t \in [T]}$, generated by Algorithm* 1 *satisfy*

$$\forall z \in \mathcal{Z} \Rightarrow \frac{1}{\Gamma_T} \sum_{t=1}^{T} \langle \gamma_t F(\widehat{z}_t),\, \widehat{z}_t - z\rangle \leqslant \frac{16 L_p}{p!} \left(\frac{D(z,\, z_1)}{T}\right)^{(p+1)/2}. \tag{18}$$

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 (18) is smaller than ε, then $\overline{z}_t = \frac{1}{\Gamma_T} \sum\limits_{t=1}^{T} \gamma_t z_t$ is an ε-solution to regular MVI (15). Hence, it is also a solution to a convex-concave SPP. A natural way to improve the method for the convex-concave problem in a tighter strongly-convex-strongly-concave setting is to use restarts [Stonyakin et al., 2018]. As a result, we obtain Algorithm 2.

Algorithm 2. Restarted HighOrderMirrorProx

Input $z_1 \in \mathcal{Z}$, $p \geqslant 1$, $0 < \varepsilon_G < 1$, $R\colon R \geqslant \|z_1 - z^*\|_2$.
$k = 1$
$\widetilde{z}_1 = z_1$
for $i \in [n]$, where $n = \left\lceil \frac{1}{2} \log \frac{\mu R^2}{\varepsilon_G} \right\rceil$ **do**
  Set $R_i = \frac{R}{2^{i-1}}$
  Set $T_i = \left\lceil \left(\frac{64 L_p R_i^{p-1}}{\mu} \right)^{2/(p+1)} \right\rceil$
  Run Algorithm 1 with $\widetilde{z}_i$, p, T_i as input
  $\widetilde{z}_{i+1} = \overline{z}_{T_i}$
return $\widetilde{z}_i$

Theorem 1. *Suppose $F\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$, which is defined in* (11)*, is pth-order Lipschitz and μ-strongly monotone* (*conditions* 1 *and* 4 *hold*)*. Denote R such that $R \geqslant \|z_1 - z^*\|_2$. Then Algorithm* 2

complexity is

$$O\left(\left(\frac{L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \log \frac{\mu R^2}{\varepsilon_G}\right). \tag{19}$$

Proof. From (17) and (18) we get the following:

$$\sum_{t=1}^{T} \gamma_t \langle F(\widehat{z}_t) - F(z^*); \widehat{z}_t - z^* \rangle \leqslant \frac{16 L_p}{p!} \left(\frac{\|z_1 - z^*\|_2^2}{2T}\right)^{(p+1)/2}. \tag{20}$$

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

$$\mu \|\overline{z}_T - z^*\|_2^2 \overset{(*)}{\leqslant} \frac{\mu}{\Gamma_T} \sum_{t=1}^{T} \gamma_t \|\widehat{z}_t - z^*\|_2^2 \overset{(10)}{\leqslant} \frac{1}{\Gamma_T} \sum_{t=1}^{T} \gamma_t \langle F(\widehat{z}_t) - F(z^*); \widehat{z}_t - z^* \rangle \overset{(20)}{\leqslant}$$

$$\overset{(20)}{\leqslant} \frac{16 L_p}{p!} \left(\frac{\|z_1 - z^*\|_2^2}{2T}\right)^{(p+1)/2}, \tag{21}$$

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

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

$$\mu \|\overline{z}_{T_i} - z^*\|_2^2 \overset{(21)}{\leqslant} \frac{16 L_p}{p!} \left(\frac{\|\widetilde{z}_i - z^*\|_2^2}{2T_i}\right)^{(p+1)/2} \leqslant \frac{16 L_p}{p!} \left(\frac{R_i^2}{2T_i}\right)^{(p+1)/2} \leqslant \frac{\mu \|\widetilde{z}_i - z^*\|_2^2}{4} \leqslant \frac{\mu R_i^2}{4} \Leftrightarrow$$

$$\Leftrightarrow T_i \geqslant \frac{R_i^2}{2} \left(\frac{64 L_p}{p! \mu R_i^2}\right)^{2/(p+1)} \geqslant \left(\frac{64 L_p R_i^{p-1}}{\mu}\right)^{2/(p+1)} = \left\lceil \left(\frac{64 L_p R_i^{p-1}}{\mu}\right)^{2/(p+1)} \right\rceil.$$

Next we need to obtain the number of restarts n required to achieve the desired accuracy. From (20) we get

$$\frac{1}{\Gamma_{T_n}} \sum_{t=1}^{T_n} \gamma_t \langle F(\widehat{z}_t) - F(z^*); \widehat{z}_t - z^* \rangle \leqslant \frac{16 L_p}{p!} \left(\frac{\|\widetilde{z}_n - z^*\|_2^2}{2T_n}\right)^{(p+1)/2} \leqslant 16 L_p \left(\frac{R_n^2}{\left(\frac{64 L_p R_n^{p-1}}{\mu}\right)^{2/(p+1)}}\right)^{(p+1)/2} =$$

$$= \frac{\mu R_n^2}{4} = \frac{\mu R^2}{2^{2n}} \leqslant \varepsilon_G \Leftrightarrow n \geqslant \frac{1}{2} \log \frac{\mu R^2}{\varepsilon_G} = \left\lceil \frac{1}{2} \log \frac{\mu R^2}{\varepsilon_G} \right\rceil.$$

Finally, the total number of iterations is

$$N = \sum_{i=1}^{n} T_i = \sum_{i=1}^{n} \left\lceil \left(\frac{64 L_p R_i^{p-1}}{\mu}\right)^{2/(p+1)} \right\rceil \leqslant \left(\frac{64 L_p}{\mu}\right)^{2/(p+1)} \sum_{i=1}^{n} R_i^{2(p-1)/(p+1)} + n \leqslant$$

$$\leqslant \left(\frac{64 L_p R^{p-1}}{\mu}\right)^{2/(p+1)} n + n = \left(\frac{64 L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \left\lceil \frac{1}{2} \log \frac{\mu R^2}{\varepsilon_G} \right\rceil + \left\lceil \frac{1}{2} \log \frac{\mu R^2}{\varepsilon_G} \right\rceil =$$

$$= O\left(\left(\frac{L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \log \frac{\mu R^2}{\varepsilon_G}\right).$$

This completes the proof. □

Local quadratic convergence

Just as in the previous subsection, an addition to introducing Algorithm 3 and its convergence rate, we need to provide some prerequisite information from [Huang, Zhang, Zhang, 2020].

Algorithm 3. CRN-SPP [Algorithm 1 in [Huang, Zhang, Zhang, 2020]]

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

Because of the strong convexity and the 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 [Huang, Zhang, Zhang, 2020] in the analysis of Algorithm 3 complexity.

$$m(z) := \frac{1}{2}\|F(z)\|_2^2 = \frac{1}{2}(\|\nabla_x g(x, y)\|_2^2 + \|\nabla_y g(x, y)\|_2^2). \tag{22}$$

Algorithm 3 solves an additional saddle point subproblem on each step, which 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 conditions 1 and 2.

Proposal 1 (Proposition 2.5 from [Huang, Zhang, Zhang, 2020]). *Let conditions* 1 *and* 2 *hold. For problem* (1) *and any point* $z = (x, y)$ *the duality gap* (5) *and the merit function* (22) *satisfy the following inequalities*

$$\frac{\mu}{L_1^2}m(z) \leqslant G(x, y) \leqslant \frac{L_1}{\mu^2}m(z). \tag{23}$$

The next theorem proves the local quadratic convergence of Algorithm 3, and it is based on Theorem 3.6 from [Huang, Zhang, Zhang, 2020].

Theorem 2 (Theorem 3.6 from [Huang, Zhang, Zhang, 2020]). *Suppose* $F\colon \mathcal{Z} \to \mathbb{R}^n$ *is a* μ*-strongly monotone, first- and second-order Lipschitz operator* (*conditions* 1, 2 *and* 3 *hold). Let* $\{z_k\}$ *be generated by Algorithm* 3 *with* $\overline{\gamma} = \frac{L_2\mu^2}{2L^2}$, $\xi = \max\left\{1, \frac{L_1}{\mu}\right\}$ *and*

$$z_0\colon \|z_0 - z^*\|_2 \leqslant \frac{\mu}{L_2\xi}. \tag{24}$$

Then

$$\forall k \geqslant 0\ \|z_{k+1} - z^*\|_2 \leqslant \frac{L_2\xi}{\mu}\|z_k - z^*\|_2^2. \tag{25}$$

Proof. Here we provide only the modified part of its proof. The rest of it can be found in [Huang, Zhang, Zhang, 2020].

If $z_{k+1} = \widetilde{z}_{k+1} = z_k + d_k$, then

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

Otherwise, if $z_{k+1} = \widehat{z}_{k+1} = z_k + \alpha d_k$, then

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

Hence, we get (25).

Now we need to find the area where (25) works:

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

Thus, we get (24). □

Our idea is to use Algorithm 2 until it reaches the area (24) and then to switch to Algorithm 3. Algorithm 4 provides the pseudocode of this idea. From Proposition 1, our Theorem 1 and Theorem 2, we obtain the complexity of Algorithm 4.

Algorithm 4. Restarted HighOrderMirrorProx with local quadratic convergence

Input $z_1 \in \mathcal{Z}$, $p \geqslant 1$, $0 < \varepsilon_G < 1$, R: $R \geqslant \|z_1 - z^*\|_2$, $\rho \in (0, 1)$, $\alpha \in (0, 1)$.
$\widetilde{z}_1 = z_1$
for $i \in [n]$, where $n = \left\lceil \log \frac{L_2R\xi}{\mu} + 1 \right\rceil$ **do**
  Set $R_i = \frac{R}{2^{i-1}}$
  Set $T_i = \left\lfloor \frac{R_i^2}{2}\left(\frac{64L_p}{p!\mu R_i}\right)^{2/(p+1)} \right\rfloor$
  Run Algorithm 1 with $\widetilde{z}_i$, p, T_i as input
  $\widetilde{z}_{i+1} = \overline{z}_{T_i}$
Run Algorithm 3 with $\widetilde{z}_{i+1}$, $\widetilde{\varepsilon} = \frac{\mu^2\varepsilon_G}{L}$, $\overline{\gamma} = \frac{L_2\mu^2}{2L_1^2}$, ρ, α, g as input
return z_k

Theorem 3. *Suppose* $F\colon \mathbb{R}^n\times\mathbb{R}^m \to \mathbb{R}^n\times\mathbb{R}^m$, *which is defined in* (11), *is a* μ*-strongly monotone, first-, second- and pth-order Lipschitz operator* (*all conditions* 1, 2, 3, 4 *hold*). *Denote* $R\colon R \geqslant \|z_1 - z^*\|_2$ *and* $\xi = \max\left\{1, \frac{L_1}{\mu}\right\}$. *Then the complexity of Algorithm* 4 *is*

$$O\left(\left(\frac{L_p R^{p-1}}{\mu}\right)^{2/(p+1)} \log\frac{L_2\xi R}{\mu} + \log\frac{\log\frac{L_1^3}{2\mu^2\varepsilon_G}}{\log\frac{L_1L_2}{\mu^2}}\right). \tag{26}$$

Proof. First of all, we need to find the number of restarts n of Algorithm 2 to reach the area of the local quadratic convergence of Algorithm 3 from (24): $\|\widetilde{z}_n - z^*\|_2 \leqslant \frac{\mu}{L_2\xi}$. We can choose such n that

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

Therefore, the number of restarts is

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

Next, we switch to Algorithm 3 and we need to obtain its number of iterations until convergence. Denote by ε' the accuracy of the solution in terms of the merit function (22). Owing to the first-order Lipschitzness of $F(z)$ and the fact that $F(z^*) = 0$, we can get

$$\varepsilon' = m(z_k) = \frac{1}{2}\|F(z_k)\|_2^2 = \frac{1}{2}\|F(z_k) - F(z^*)\|_2^2 \leqslant \frac{L_1^2}{2}\|z_k - z^*\|_2^2. \tag{27}$$

Now we establish a connection between the solution in terms of the merit function $m(z)$ and the duality gap $G(x, y)$. From (27) and (23) we get the following:

$$\varepsilon_G = G(x, y) = \max_{y'\in\mathbb{R}^n} f(x, y') - \min_{x'\in\mathbb{R}^n} f(x', y) \leqslant \frac{L_1}{\mu^2} m(z_k) = \frac{L_1}{\mu^2}\varepsilon' \Leftrightarrow \frac{\mu^2\varepsilon_G}{L_1} \leqslant \varepsilon'. \tag{28}$$

Then, from (25), (24), (27) and (28) we can obtain the needed number of iterations k

$$\begin{aligned}\frac{\mu^2\varepsilon_G}{L_1} &\overset{(27),(28)}{\leqslant} \frac{L_1^2}{2}\|z_k - z^*\|_2^2 \overset{(25)}{\leqslant} \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\|z_{k-1} - z^*\|_2^2\right)^2 \leqslant \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\left(\frac{L_1L_2}{\mu^2}\|z_{k-2} - z^*\|_2^2\right)^2\right)^2 \leqslant \ldots \leqslant \\ &\leqslant \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\right)^{2^{k-1}-2}\|z_1 - z^*\|_2^{2^k} \overset{(24)}{\leqslant} \frac{L_1^2}{2}\left(\frac{L_1L_2}{\mu^2}\right)^{2^{k-1}-2}\left(\frac{\mu^2}{L_1L_2}\right)^{2^k} \Leftrightarrow \\ &\Leftrightarrow \frac{2\mu^2\varepsilon_G}{L_1^3} \leqslant \left(\frac{\mu^2}{L_1L_2}\right)^{2^{k-1}+2} \Leftrightarrow \log\frac{2\mu^2\varepsilon_G}{L_1^3} \leqslant (2^{k-1}+2)\log\frac{\mu^2}{L_1L_2}.\end{aligned}$$

Since $\log\left(\frac{\mu^2}{L_1L_2}\right) < 0$,

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

Finally, the total number of iterations of Algorithm 4 is

$$N = \sum_{i=1}^{n} T_i + k \leqslant \left(\frac{64L_pR^{p-1}}{\mu}\right)^{2/(p+1)} \left\lceil \log \frac{L_2\xi R}{\mu} + 1 \right\rceil + \left\lceil \log \frac{L_2\xi R}{\mu} + 1 \right\rceil + \left\lceil \frac{\log \frac{L_1^3}{2\mu^2\varepsilon_G}}{\log \frac{L_1L_2}{\mu^2}} \right\rceil + 1 =$$

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

□

Gradient norm minimization

In this subsection we apply the framework of [Dvurechensky et al., 2019] to Algorithm 4, introduce Algorithm 5 for problem (7) 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. *If $g(x, y)$ is p-Lipchitz* (14)*, then its partial pth-order derivatives are also Lipschitz.*

$$\forall \widehat{x},\ x \in \mathbb{R}^n,\ \widehat{y},\ y \in \mathbb{R}^m \Rightarrow \|\nabla^p_{x^i y^{p-i}} g(\widehat{x}, \widehat{y}) - \nabla^p_{x^i y^{p-i}} g(x, y)\|_2 \leqslant L_p \|\widehat{z} - z\|_2. \tag{29}$$

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

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

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

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

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

□

Lemma 4. *Let $\nabla^p_{x\ldots x} g(x, y)$ be Lipschitz* (29)*. Then*

$$\forall n \in [p] \Rightarrow \|\nabla^{p-n}_{x\ldots x} g(\widehat{z}) - \nabla^{p-n}_{x\ldots x} \Phi_{(x,y),p}(\widehat{z})\|_2 \leqslant \frac{L_p \left(\sqrt{2}\right)^n}{(n+1)!} \|\widehat{z} - z\|_2^{n+1}. \tag{30}$$

Proof. We prove this by induction.

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

$$\|\nabla^{p-1}_{x\dots x} g(\widehat{z}) - \nabla^{p-1}_{x\dots x}\Phi_{(x,y),p}(\widehat{z})\|_2 = \|\nabla^{p-1}_{x\dots x} g(\widehat{z}) - \nabla^{p-1}_{x\dots x} g(z) - \nabla^{p}_{x\dots xx} g(z)[\widehat{x} - x] - \nabla^{p}_{x\dots xy} g(z)[\widehat{y} - y]\|_2 =$$

$$= \|f(\widehat{z}) - f(z) - \nabla f(z)[\widehat{z} - z]\|_2 = \left\| \int_0^1 \langle \nabla f(z + \tau(\widehat{z} - z)) - \nabla f(z); \widehat{z} - z\rangle \, d\tau \right\|_2 \leqslant$$

$$\leqslant \int_0^1 \left\| \begin{pmatrix} \nabla^{p}_{x\dots xx} g(z + \tau(\widehat{z} - z)) \\ \nabla^{p}_{x\dots xy} g(z + \tau(\widehat{z} - z)) \end{pmatrix} - \begin{pmatrix} \nabla^{p}_{x\dots xx} g(z) \\ \nabla^{p}_{x\dots xy} g(z) \end{pmatrix} \right\|_2 \|\widehat{z} - z\|_2 \, d\tau =$$

$$= \int_0^1 \sqrt{\|\nabla^{p}_{x\dots xx} g(z + \tau(\widehat{z} - z)) - \nabla^{p}_{x\dots xx} g(z)\|_2^2 + \|\nabla^{p}_{x\dots xy} g(z + \tau(\widehat{z} - z)) - \nabla^{p}_{x\dots xy} g(z)\|_2^2} \|\widehat{z} - z\|_2 \, d\tau \overset{(29)}{\leqslant}$$

$$\overset{(29)}{\leqslant} \sqrt{2} L_p \|\widehat{z} - z\|_2^2 \int_0^1 \tau \, d\tau = \frac{L_p \sqrt{2}}{2} \|\widehat{z} - z\|_2^2.$$

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

$$\|\nabla_x g(\widehat{z}) - \nabla_x \Phi_{(x,y),p}(\widehat{z})\|_2 = \Big\| \nabla_x g(\widehat{z}) - \nabla_x g(z) - (\nabla^2_{xx} g(z)[\widehat{x} - x] - \nabla^2_{xy} g(z)[\widehat{y} - y]) - \dots -$$

$$- \nabla_x \left(\frac{1}{p!} \nabla^p g(z)[\widehat{z} - z]^p \right) \Big\|_2 \leqslant \frac{L_p \left(\sqrt{2}\right)^{p-1}}{p!} \|\widehat{z} - z\|_2^p. \quad (31)$$

And consider $n = p$

$$|g(\widehat{z}) - \Phi_{(x,y),p}(\widehat{z})| = \left| g(\widehat{z}) - g(z) - \nabla_x g(z)[\widehat{x} - x] - \nabla_y g(z)[\widehat{y} - y] - \dots - \frac{1}{p!} \nabla^p g(z)[\widehat{z} - z]^p \right| \leqslant$$

$$\leqslant \int_0^1 \left\| \begin{pmatrix} \nabla_x g(z + \tau(\widehat{z} - z)) \\ \nabla_y g(z + \tau(\widehat{z} - z)) \end{pmatrix} - \begin{pmatrix} \nabla_x g(z) \\ \nabla_y g(z) \end{pmatrix} - \tau \begin{pmatrix} \nabla^2_{xx} g(z)[\widehat{x} - x] + \nabla^2_{xy} g(z)[\widehat{y} - y] \\ \nabla^2_{yx} g(z)[\widehat{x} - x] + \nabla^2_{yy} g(z)[\widehat{y} - y] \end{pmatrix} - \dots -$$

$$- \frac{\tau^{p-1}}{p!} \begin{pmatrix} \nabla_x (\nabla^p g(z)[\widehat{z} - z]^p) \\ \nabla_y (\nabla^p g(z)[\widehat{z} - z]^p) \end{pmatrix} \Big\|_2 \|\widehat{z} - z\|_2 \, d\tau = \int_0^1 \Big(\Big\| \nabla_x g(z + \tau(\widehat{z} - z)) - \nabla_x g(z) -$$

$$- \tau(\nabla^2_{xx} g(z)[\widehat{x} - x] + \nabla^2_{xy} g(z)[\widehat{y} - y]) - \dots - \frac{\tau^{p-1}}{p!} \nabla_x (\nabla^p g(z)[\widehat{z} - z]^p) \Big\|_2^2 +$$

$$+ \Big\| \nabla_y g(z + \tau(\widehat{z} - z)) - \nabla_y g(z) - \tau(\nabla^2_{yx} g(z)[\widehat{x} - x] + \nabla^2_{yy} g(z)[\widehat{y} - y]) - \dots -$$

$$- \frac{\tau^{p-1}}{p!} \nabla_y (\nabla^p g(z)[\widehat{z} - z]^p) \Big\|_2^2 \Big)^{1/2} \|\widehat{z} - z\|_2 \, d\tau.$$

If we denote $\widehat{z} = z + \tau(\widehat{z} - z)$ in (31), each of two factors under the square root is indeed what we had for $n = p - 1$. Finally,

$$\|\nabla_x g(\widehat{z}) - \nabla_x \Phi_{(x,y),p}(\widehat{z})\|_2 \leqslant \sqrt{2} \frac{L_p \left(\sqrt{2}\right)^{p-1}}{p!} \|\widehat{z} - z\|_2^{p+1} \int_0^1 \tau^p \, d\tau = \frac{L_p \left(\sqrt{2}\right)^{p}}{(p+1)!} \|\widehat{z} - z\|_2^{p+1}.$$

For any other partial derivative in (30) 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 [Grapiglia, Nesterov, 2019] for SPP.

Lemma 5 (Lemma 5.2 from [Grapiglia, Nesterov, 2019]). *Let* $(\widetilde{x}, \widetilde{y}) = T^{g}_{p,M}(x, y)$, $p \geqslant 2$, *where* $M \geqslant \sqrt{2}pL_p > \frac{1}{\sqrt{2}}pL_p$ *and condition* 4 *holds. Then*

$$\|\nabla g(\widetilde{x}, \widetilde{y})\|_2^{(p+1)/p} \frac{M^{(3p+1)/(2p)}}{2^{(2p^2+p+1)/(2p)} p(p+1)!} \leqslant g(x, \widetilde{y}) - g(\widetilde{x}, y). \tag{32}$$

Proof.

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

Firstly, consider ∇_x:

$$\|\nabla_x g(\widetilde{x}, \widetilde{y})\|_2^2 = \|\nabla_x g(\widetilde{x}, \widetilde{y}) - \nabla_x \Phi_{(x,y),p}(\widetilde{x}, \widetilde{y}) + \nabla_x \Phi_{(x,y),p}(\widetilde{x}, \widetilde{y}) - \nabla_x \Omega_{(x,y),p,M}(\widetilde{x}, \widetilde{y}) + \nabla_x \Omega_{(x,y),p,M}(\widetilde{x}, \widetilde{y})\|_2^2 \leqslant$$
$$\leqslant \left(\|\nabla_x g(\widetilde{x}, \widetilde{y}) - \nabla_x \Phi_{(x,y),p}(\widetilde{x}, \widetilde{y})\|_2 + \|\nabla_x \Phi_{(x,y),p}(\widetilde{x}, \widetilde{y}) - \nabla_x \Omega_{(x,y),p,M}(\widetilde{x}, \widetilde{y})\| + \|\nabla_x \Omega_{(x,y),p,M}(\widetilde{x}, \widetilde{y})\|_2\right)^2 \leqslant$$
$$\leqslant \left(\frac{2^{(p-1)/2} L_p}{p!}\|\widetilde{z} - z\|_2^p + \frac{2^{(p-1)/2} M}{p!}\|\widetilde{x} - x\|_2^p\right)^2 \leqslant 2^p M^2 \|\widetilde{z} - z\|_2^{2p}.$$

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

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

Summing these two results, we obtain

$$\|\nabla g(\widetilde{x}, \widetilde{y})\|_2^2 \leqslant 2^{p+1} M \left(\|\widetilde{x} - x\|_2^2 + \|\widetilde{y} - y\|_2^2\right)^p. \tag{33}$$

Secondly, consider point $(\widetilde{x}, y)$. From (30) it is obvious that

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

From this fact we get

$$g(\widetilde{x}, y) \leqslant \Phi_{(x,y),p}(\widetilde{x}, y) + \frac{L_p\left(\sqrt{2}\right)^p}{(p+1)!}\|\widetilde{x} - x\|_2^{p+1} = \Phi_{(x,y),p}(\widetilde{x}, y) + \frac{L_p\left(\sqrt{2}\right)^{p-1}}{(p+1)!}\|\widetilde{x} - x\|_2^{p+1} -$$
$$- \left(\frac{M\left(\sqrt{2}\right)^{p-1}}{(p+1)!}\|\widetilde{x} - x\|_2^{p+1} - \frac{L_p\left(\sqrt{2}\right)^p}{(p+1)!}\|\widetilde{x} - x\|_2^{p+1}\right) = \Omega_{(x,y),p,M}(\widetilde{x}, y) - \left(M - L_p\sqrt{2}\right)\frac{\left(\sqrt{2}\right)^{p-1}\|\widetilde{x} - x\|_2^{p+1}}{(p+1)!} \leqslant$$
$$\leqslant \Omega_{(x,y),p,M}(\widetilde{x}, \widetilde{y}) - \left(M - L_p\sqrt{2}\right)\frac{\left(\sqrt{2}\right)^{p-1}\|\widetilde{x} - x\|_2^{p+1}}{(p+1)!}.$$

Since $M \geqslant \sqrt{2}pL_p \Leftrightarrow -L_p\sqrt{2} \geqslant -\frac{M}{p}$. We have

$$\Omega_{(x,y),p,M}(\widetilde{x}, \widetilde{y}) - g(\widetilde{x}, y) \geqslant \frac{M(p-1)\left(\sqrt{2}\right)^{p-1}\|\widetilde{x} - x\|_2^{p+1}}{p(p+1)!} \geqslant \frac{M\|\widetilde{x} - x\|_2^{p+1}}{p(p+1)!}. \tag{34}$$

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

$$g(x, \widetilde{y}) - \Omega_{(x,y),p,M}(\widetilde{x}, \widetilde{y}) \geqslant \frac{M\|\widetilde{y} - y\|_2^{p+1}}{p(p+1)!}. \tag{35}$$

From the sum of (34) and (35) we obtain

$$g(x, \widetilde{y}) - g(\widetilde{x}, y) \geqslant \frac{M}{p(p+1)!}\left(\|\widetilde{x} - x\|_2^{p+1} + \|\widetilde{y} - y\|_2^{p+1}\right). \tag{36}$$

Finally, we need to connect (33) and (36). From Hölder's inequality we can get

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

where $q,\ p \in \mathbb{N},\ q > p \geqslant 1$. Now from (33) it follows that

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

And from (36) we can get

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

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

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

□

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

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

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

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

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 $T_i = \left\lceil \left(\frac{64L_pR_i^{p-1}}{p!\mu}\right)^{2/(p+1)} \right\rceil$
  Run Algorithm 1 for g_μ with $\widetilde{z}_i$, p, T_i as input
  $\widetilde{z}_{i+1} = \overline{z}_{T_i}$
Run Algorithm 3 with $\widetilde{z}_{i+1}$, ε', $\overline{\gamma} = \frac{L_2\mu^2}{2L_1^2}$, ρ, α, g_μ as input
Find $\widetilde{z} = T_{p,M}^{g_\mu}(z_k)$
Output $\widetilde{z}$.

Theorem 4. *Assume the function $g(x, y)\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is convex by x and concave by y, p times differentiable on $\mathbb{R}^n$ with L_p-Lipschitz pth derivative. Let $\widetilde{z}$ be generated by Algorithm 5. Then*

$$\|\nabla g(\widetilde{z})\|_2 \leqslant \varepsilon_\nabla,$$

and the total complexity of Algorithm 5 is

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

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

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

$$\begin{aligned}
g(x_\mu^*, y_1) + \frac{\mu}{2}\|x_\mu^* - x_1\|_2^2 = g_\mu(x_\mu^*, y_1) \leqslant g_\mu(x^*, y_1) = g(x^*, y_1) + \frac{\mu}{2}\|x^* - x_1\|_2^2 \leqslant \\
\leqslant g(x_\mu^*, y_1) + \frac{\mu}{2}\|x^* - x_1\|_2^2 \Leftrightarrow \|x_\mu^* - x_1\|_2 \leqslant \|x^* - x_1\|_2; \\
g(x_1, y_\mu^*) - \frac{\mu}{2}\|y_\mu^* - y_1\|_2 = g_\mu(x_1, y_\mu^*) \geqslant g_\mu(x_1, y^*) = g(x_1, y^*) - \frac{\mu}{2}\|y^* - y_1\|_2^2 \geqslant \\
\geqslant g(x_1, y_\mu^*) - \frac{\mu}{2}\|y^* - y_1\|_2^2 \Leftrightarrow \|y_\mu^* - y_1\|_2 \leqslant \|y^* - y_1\|_2.
\end{aligned}$$

This gives us

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

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

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

From Theorem 3 we already know the number of restarts to reach the area of quadratic convergence: $n = \left\lceil \log \frac{L_2 R \xi}{\mu} + 1 \right\rceil$.

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

$$\begin{aligned}
\|\nabla g(\widetilde{x}, \widetilde{y})\|_2^2 = \|\nabla_x g_\mu(\widetilde{x}, \widetilde{y}) - \mu(\widetilde{x} - x_1)\|_2^2 + \|\nabla_y g_\mu(\widetilde{x}, \widetilde{y}) + \mu(\widetilde{y} - y_1)\|_2^2 \leqslant \\
\leqslant \left(\|\nabla_x g_\mu(\widetilde{x}, \widetilde{y})\|_2 + \mu\|\widetilde{x} - x\|_2\right)^2 + \left(\|\nabla_y g_\mu(\widetilde{x}, \widetilde{y})\|_2 + \mu\|\widetilde{y} - y\|_2\right)^2 \leqslant \\
\leqslant 2\left(\|\nabla_x g_\mu(\widetilde{x}, \widetilde{y})\|_2^2 + \|\nabla_y g_\mu(\widetilde{x}, \widetilde{y})\|_2^2\right) + 2\mu^2\left(\|\widetilde{x} - x\|_2^2 + \|\widetilde{y} - y\|_2^2\right) = \\
= 2\|\nabla g_\mu(\widetilde{x}, \widetilde{y})\|_2^2 + 2\mu^2\|\widetilde{z} - z_1\|_2^2 \Leftrightarrow \|\nabla g(\widetilde{x}, \widetilde{y})\|_2 \leqslant \sqrt{2\|\nabla g_\mu(\widetilde{x}, \widetilde{y})\|_2^2 + 2\mu^2\|\widetilde{z} - z_1\|_2^2}.
\end{aligned}$$

Firstly, we estimate $\|\nabla g_\mu(\widetilde{x}, \widetilde{y})\|_2$. From (32) we know that

$$\|\nabla g_\mu(\widetilde{x}, \widetilde{y})\|_2^{(p+1)/p} \frac{M^{(3p+1)/(2p)}}{2^{(2p^2+p+1)/(2p)} p(p+1)!} \overset{(32)}{\leqslant} g_\mu(x, \widetilde{y}) - g_\mu(\widetilde{x}, y) \leqslant$$

$$\leqslant \max_{\widetilde{y} \in \mathbb{R}^m} g_\mu(x, \widetilde{y}) - \min_{\widetilde{x} \in \mathbb{R}^n} g_\mu(\widetilde{x}, y) = G_\mu(x, y) \leqslant \varepsilon' \Leftrightarrow$$

$$\Leftrightarrow \|\nabla g_\mu(\widetilde{x}, \widetilde{y})\|_2 \leqslant \left(\frac{2^{(2p^2+p+1)/(2p)} p(p+1)! \varepsilon'}{M^{(3p+1)/(2p)}} \right)^{p/(p+1)} = \frac{\varepsilon_\nabla}{2}. \quad (37)$$

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

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

And since $\widetilde{z}$ is closer to the solution, z_1, we have

$$\|\widetilde{z} - z^*\|_2 \leqslant \|z^* - z_1\|2 \leqslant R.$$

From these facts and the triangle inequality we get

$$\mu\|\widetilde{z} - z_1\|_2 \leqslant \mu \left(\|\widetilde{z} - z^*\|_2 + \|z^* - z_1\|_2 \right) \leqslant 2R\mu = \frac{\varepsilon_\nabla}{2}. \quad (38)$$

Thus, from (37) and (38) we obtain

$$\|\nabla g_\mu(\widetilde{x}, \widetilde{y})\|_2 \leqslant \sqrt{\frac{2\varepsilon_\nabla^2}{4} + \frac{2\varepsilon_\nabla^2}{4}} = \varepsilon_\nabla.$$

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

$$N = \sum_{i=1}^n T_i + k \leqslant \left(\frac{64 L_p}{p! \mu} \right)^{2/(p+1)} \sum_{i=1}^n R_i^{2(p-1)/(p+1)} + n + k \leqslant$$

$$\leqslant \left(\frac{64 L_p R^{p-1}}{p! \mu} \right)^{2/(p+1)} \cdot n + n + k = O\left(\left(\frac{L_p R^p}{\varepsilon_\nabla} \right)^{2/(p+1)} \log \frac{L_2 R^2 \xi}{\varepsilon_\nabla} \right),$$

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

Discussion

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

The methods for the minimax problem are based on the ideas developed in [Bullins, Lai, 2020; Huang, Zhang, Zhang, 2020]. In [Bullins, Lai, 2020] the authors use pth-order oracle to construct an algorithm for MVI problems with a monotone operator. As a corollary, this algorithm allows one to solve SPP with a convex-concave objective. Because of the strong convexity and the strong concavity of our problem, we can apply a restart technique to the method from [Bullins, Lai, 2020] and get a better algorithm complexity. To further improve the local convergence rate, we switch to the algorithm from [Huang, Zhang, Zhang, 2020] in the area of its quadratic convergence. In this way we get rid of

the multiplicative logarithmic factor and get an additive log log factor in the final complexity estimate and get a locally quadratic convergence.

The method for gradient norm minimization relies on the works [Grapiglia, Nesterov, 2019] and [Dvurechensky et al., 2019]. From [Grapiglia, Nesterov, 2019] we take the result that connects the 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 [Dvurechensky et al., 2019] and use our optimal algorithm for minimax SPP for gradient norm minimization.

In spite of all the improvements, we should recall additional conditions about the problem, which reduces the number of real problems, that can suit it.

One of possible directions for further research are the more general Hölder conditions instead of Lipschitz conditions and the uniformly convex case. Additionally, the author in [Bullins, Lai, 2020] provided implementation details of Algorithm 1 only for $p = 2$. Therefore, the questions of its realizaition for $p > 2$ are still open.

References

Bullins B. On lower iteration complexity bounds for the saddle point problems // arXiv preprint. — 2018. — https://arxiv.org/pdf/1812.10349

Bullins B., Lai K. A. Higher-order methods for convex-concave min-max optimization and monotone variational inequalities // arXiv preprint. — 2020. — https://arxiv.org/pdf/2007.04528

Bullins B., Peng R. Higher-Order Accelerated Methods for Faster Non-Smooth Optimization // arXiv preprint. — 2019. — https://arxiv.org/pdf/1906.01621

Dvurechensky P., Gasnikov A., Ostroukhov P., Uribe C. A., Ivanova A. Near-optimal tensor methods for minimizing the gradient norm of convex function // arXiv preprint. — 2019. — https://arxiv.org/pdf/1912.03381

Gasnikov A., Dvinskikh D., Dvurechensky P., Kamzolov D., Pasechnyk D., Matykhin V., Tupitsa N., Chernov A. Accelerated meta-algorithm for convex optimization // Computational Mathematics and Mathematical Physics. — 2020. — Vol. 61, No. 1.

Gasnikov A., Dvurechensky P., Gorbunov E., Vorontsova E., Selikhanovych D., Uribe C. A. Optimal Tensor Methods in Smooth Convex and Uniformly Convex Optimization // Proceedings of the Thirty-Second Conference on Learning Theory. — 2019. — Vol. 99. — P. 1374–1391. — http://proceedings.mlr.press/v99/gasnikov19a/gasnikov19a.pdf

Gasnikov A., Dvurechensky P., Gorbunov E., Vorontsova E., Selikhanovych D., Uribe C. A., Jiang B., Wang H., Zhang S., Bubeck S., Jiang Q., Lee Y. T., Li Y., Sidford A. Near Optimal Methods for Minimizing Convex Functions with Lipschitz p-th Derivatives // Proceedings of the Thirty-Second Conference on Learning Theory. — 2019. — Vol. 99. — P. 1392–1393. — http://proceedings.mlr.press/v99/gasnikov19b/gasnikov19b.pdf

Gidel G., Berard H., Vignoud G., Vincent P., Lacoste-Julien S. A variational inequality perspective on generative adversarial networks // arXiv preprint. — 2018. — https://arxiv.org/pdf/1802.10551

Grapiglia G. N., Nesterov Yu. High-Order Oracle Complexity of Smooth and Strongly Convex Optimization // arXiv preprint. — 2019. — https://arxiv.org/pdf/1907.07053

Hoffmann K. H., Kornstaedt H. J. Higher-order necessary conditions in abstract mathematical programming // Journal of Optimization Theory and Applications. — 1978. — Vol. 26, No. 4. — P. 533–568. — https://doi.org/10.1007/BF00933151

Huang K., Zhang J., Zhang S. Cubic Regularized Newton Method for Saddle Point Models: a Global and Local Convergence Analysis // arXiv preprint. — 2020. — https://arxiv.org/pdf/2008.09919

Kornowski G., Shamir O. High-Order Oracle Complexity of Smooth and Strongly Convex Optimization // arXiv preprint. — 2020. — https://arxiv.org/pdf/2010.06642

Korpelevich G. The extragradient method for finding saddle points and other problems // Eknomika i Matematicheskie Metody. — 1976. — Vol. 12. — P. 747–756.

Lin T., Jin C., Jordan M. et al. Near-optimal algorithms for minimax optimization // arXiv preprint. — 2020. — https://arxiv.org/pdf/2002.02417

Mokhtari A., Ozdaglar A., Pattathil S. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach // International Conference on Artificial Intelligence and Statistics. — PMLR, 2020. — P. 1497–1507.

Monteiro R. D. C., Svaiter B. F. Iteration-complexity of a Newton proximal extragradient method for monotone variational inequalities and inclusion problems // SIAM Journal on Optimization. — 2012. — Vol. 22, No. 3. — P. 914–935.

Monteiro R. D. C., Svaiter B. F. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean // SIAM Journal on Optimization. — 2010. — Vol. 20, No. 6. — P. 2755–2787.

Nemirovski A. 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. — 2004. — Vol. 15, No. 1. — P. 229–251.

Nemirovsky A. S., Yudin D. B. Problem Complexity and Method Efficiency in Optimization. — New York: J. Wiley & Sons, 1983.

Nesterov Yu. Dual extrapolation and its applications to solving variational inequalities and related problems // Mathematical Programming. — 2007. — Vol. 109, No. 2-3. — P. 319–344.

Nesterov Yu., Nemirovskii A. Interior-point polynomial algorithms in convex programming. — Vol. 13. — SIAM, 1994.

Nesterov Yu., Polyak B. Cubic regularization of Newton method and its global performance // Mathematical Programming. — 2006. — Vol. 108, No. 1. — P. 177–205. — http://dx.doi.org/10.1007/s10107-006-0706-8

Nesterov Yu., Scrimali L. Solving strongly monotone variational and quasi-variational inequalities // Available at SSRN 970903. — 2006.

Nocedal J., Wright S. Numerical optimization. — Springer Science & Business Media, 2006.

Rockafellar R. T. Monotone operators and the proximal point algorithm // SIAM journal on control and optimization. — 1976. — Vol. 14, No. 5. — P. 877–898.

Stonyakin F., Gasnikov A., Dvurechensky P., Alkousa M., Titov A. Generalized Mirror Prox for Monotone Variational Inequalities: Universality and Inexact Oracle // arXiv preprint. — 2018. — https://arxiv.org/pdf/1806.05140

Tseng P. On accelerated proximal gradient methods for convex-concave optimization. — MIT, 2008. — http://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf

Tseng P. On linear convergence of iterative methods for the variational inequality problem // Journal of Computational and Applied Mathematics. — 1995. — Vol. 60, No. 1-2. — P. 237–252.

Zhang J., Hong M., Zhang S. On lower iteration complexity bounds for the saddle point problems // arXiv preprint. — 2019. — https://arxiv.org/pdf/1912.07481