Noname manuscript No.
(will be inserted by the editor)

Composite optimization models via proximal gradient method with a novel enhanced adaptive stepsizes

Pham Thi Hoai[1] · **Nguyen Pham Duy Thai**[1]

Abstract We first consider the convex *composite optimization models* with the *local Lipschitzness* condition imposed on the gradient of the differentiable term. The classical *proximal gradient method* will be studied with our novel *enhanced adaptive* stepsize selection. To obtain the convergence of the proposed algorithm, we establish a *sufficient decrease type inequality* associated with our new stepsize choice. This allows us to demonstrate the descent of the objective value from some fixed iteration and yield the *sublinear convergence rate* of the new method. Especially, in the case of *locally strong convexity* of the smooth term, our algorithm converges *Q-linearly*. We also further show that our method can be applied to *nonconvex* composite optimization problems provided that the differentiable function has a globally Lipschitz gradient. Finally, the efficiency of our proposed algorithms is shown by numerical results for numerous applicable test instances in comparison with the other state-of-the-art algorithms.

Keywords proximal gradient method · nonlinear programming · composite optimization model · locally Lipschitz gradient · lasso problem

Mathematics Subject Classification (2010) 49J40 · 47H04 · 47H10

1 Introduction

1.1 Problem description and motivation

Composite optimization models (COM) have arisen from many real-life applications, such as machine learning, signal processing, data science, etc, and have received a lot of attention recently, see e.g., [1,5,6,3,4,7,22,17,29,21,9]. The formulation of (COM) considered in this paper can be described as follows:

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + g(x), \tag{P}$$

where f and g are functions satisfying *Assumption 1* below.

Assumption 1 *(A1)* $g : \mathbb{R}^n \to (-\infty, +\infty]$ *is a proper and closed convex function.*

(A2) $f : \mathbb{R}^n \to (-\infty, +\infty]$ *is proper and closed such that* $dom(f)$ *is convex,* $dom(g) \subset int(dom(f))$ *and* f *is differentiable on* $int(dom(f))$.

✉ Pham Thi Hoai
hoai.phamthi@hust.edu.vn
Nguyen Pham Duy Thai
thai.pdn@gmail.com

[1] Faculty of Mathematics and Informatics, Hanoi University of Science and Technology, 1 Dai Co Viet Road, Hanoi, Vietnam

(A3) The optimal solution set X^* of (P) is nonempty and F_* stands for the optimal value of (P).

One of the conventional methods for solving the problem (P) is *proximal gradient method* (PG) introduced by Fukushima and Mine [15] in 1981 and has now become classical. As a matter of fact, the further origin of the proximal gradient method can be traced back to 1970s with the work of Brucks [11] and Passty [26] in the more general setting of forward backward splitting method. The detailed methodology of the PG method can be found in [4,7]. It is observed that the optimal condition for the problem (P) relates to the concept of its stationary points. Specifically, if $x^* \in \text{int}(\text{dom}(f))$ is a local optimal solution of (P) then it should be a *stationary point* of (P) (see e.g., Theorem 3.72 [4]), i.e., for some $t > 0$

$$x^* = \text{Prox}_{tg}(x^* - t\nabla f(x^*)), \text{ (see e.g., Theorem 10.7 [4])}, \tag{1.1}$$

where $\text{Prox}_{tg}(.)$ is the proximal operator and is defined as the unique optimal solution of the minimization problem

$$\text{Prox}_{tg}(y) = \text{argmin}_{x\in\mathbb{R}^n}\left\{g(x) + \frac{1}{2t}\|x-y\|^2\right\}. \tag{1.2}$$

In the convex situation of (P), i.e., f is convex, the set of stationary points of (P) is coincident with X^*. One can see [4] (Theorems 3.72 and 10.7) for more details. Based on the stationary condition (1.1), starting from some $x^0 \in \text{int}(\text{dom}(f))$, the well-known PG method to solve problem (P) is designed by generating the sequence $\{x^k\}$ according to the rule

$$x^{k+1} = \text{Prox}_{t_k g}(x^k - t_k\nabla f(x^k)), \;\; k = 0,1,2,... \tag{1.3}$$

The PG scheme (1.3) is useful if we can compute $\text{Prox}_{t_k g}(x^k - t_k\nabla f(x^k))$ easily by some explicit formulas. Though, there is a list of functions whose *proximal operator* is analytically computable and that list can be found in [4]; for instances, g is the ℓ_1 norm or the indicator function of a closed convex set $C \subset \mathbb{R}^n$. In formula (1.3), $t_k > 0, k = 0,1,2,...$ are defined as *stepsizes* which play a crucial role in the proximal gradient scheme. A suitable stepsize selection can be drawn in the two main points: firstly, it should guarantee the convergence of $\{x^k\}$ to some stationary point of problem (P); secondly, it should navigate x^k to a "good" stationary point. i.e., providing, for example, the low objective value as much as possible with a cheap computational cost. For the class of L_f- smooth function f, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \le L_f\|x-y\|, \; \forall x,y \in \text{int}(\text{dom}(f)),$$

the stepsize t_k in (1.3) can be controlled flexibly by using *constant stepsize* in $\left(0, \frac{2}{L_f}\right)$ or *backtracking line search* rule. Followed by [4] (Theorem 10.21), one gets the *sublinear rate* of convergence, i.e., computational complexity $O(\frac{1}{k})$ of $F(x^k) - F_*$ if f is assumed to be convex and t_k is either in $\left(0, \frac{1}{L_f}\right]$ or taken by backtracking procedure. In the case where f is strongly convex, the convergence rate of $\{x^k\}$ to some $x^* \in X^*$ is proved to be Q-linear. These properties can be seen as the generalization of the convergence results for the gradient descent method solving unconstrained nonlinear optimization problems, i.e., problem (P) with $g = 0$.

Recently, researchers have been concerned about problem (P) *without the global Lipschitzness assumption on* ∇f, see, e.g., [2,8,17,18,25] because the class of such functions occurs in many applied problems (see e.g., [18,30] and the references therein). In 2017, Bauschke et al. [2] proposed *NoLips Algorithm* that requires Bregman distances-based computation and constant L in the *Lipschitz-like/convexity condition* (LC). One can see [28] to find the role of non-Euclidean proximal distances of Bregman type in the development and analysis of some typical first order optimization algorithms. If the stepsize is chosen in $(0, \frac{2-\delta}{L})$ then Nolips algorithm is shown in [2] to have the convergent results similar to the ones of the normal PG scheme. Following that, Dragomir et

al. [13] give a lower bound to prove that the $O\left(\frac{1}{k}\right)$ convergence rate of the NoLips method is optimal for the class of problems satisfying the relative smoothness assumption. Nevertheless, one knows that there are some restrictions of taking stepsize within $\left(0, \frac{2}{L_f}\right)$ or $\left(0, \frac{2-\delta}{L}\right)$ like: firstly, the process of finding these constants is not easy in general and secondly, if the coefficients L_f or L are large then the constant stepsizes will be very small and that may take long executing time. Another class of ideas to resolve the lack of a globally Lipschitz gradient was proposed by Cruz and Nghia [12], Kanzow and Mehlitz [18], Jia et al. [17], and Zhao et al. [31]. While the methods in [12] and [31] are designed for convex settings, the approaches proposed in [18] and [17] can be applied to the nonconvex setting of (P) under the Kurdyka–Łojasiewicz condition. Unfortunately, their stepsize choices rely on some backtracking line search procedures, which can make each iteration computationally expensive.

To overcome the drawbacks of stepsize selection based on line search or estimating some unknown constants like L, L_f mentioned above, an interesting question should be considered is:

Question 1.1 *Under Assumption 1 and assuming that f is convex and has a locally Lipschitz gradient, is there an adaptive way to find stepsizes explicitly for PG scheme solving problem (P) such that we do not need neither estimating constants like $L_f, L, \ldots$ nor backtracking line search procedures?*

1.2 Some recent algorithms considering Question 1.1

In the specific context of the problem (P) with $g = 0$, two algorithms named AdGD and NGD were proposed by Malitsky and Mishchenko [23] (2019) and Hoai et al. [16] (2024), respectively. Both of them use explicit stepsize strategies based on the local curvature of f. In the general setting of problem (P), to give an answer to Question 1.1, Malitsky and Mishchenko [24] have developed their method AdGD [23] to AdPG (Adaptive Proximal Gradient) for solving the problem (P) recently. The stepsize of AdPG is defined by

$$t_k = t_{k-1} \min\left\{\sqrt{\frac{2}{3}+\theta_{k-1}}, \frac{1}{\sqrt{\left[\frac{2t_{k-1}^2\|\nabla f(x^k)-\nabla f(x^{k-1})\|^2}{\|x^k-x^{k-1}\|^2}-1\right]^+}}\right\}, \tag{AdPG}$$

where $\theta_0 = \frac{1}{3}, \theta_k = \frac{t_k}{t_{k-1}}, k \geq 1$. And for some $t \in \mathbb{R}$, the notation t^+ stands for $\max\{t, 0\}$. The iterates of AdPG are proved to converge to an optimal solution of (P) with the *worst-case sublinear* rate, i.e., the complexity $O(\frac{1}{k})$ of $\min\limits_{1\leq i\leq k}(F(x^i)-F_*)$. In parallel with this work, Latafat et al. [20] proposed adaPGM that has

$$t_k = t_{k-1} \min\left\{\sqrt{1+\frac{t_{k-1}}{t_{k-2}}}, \frac{1}{2\sqrt{\left[t_{k-1}\left(\frac{t_{k-1}\|\nabla f(x^k)-\nabla f(x^{k-1})\|^2-\left\langle\nabla f(x^k)-\nabla f(x^{k-1}, x^k-x^{k-1}\right\rangle}{\|x^k-x^{k-1}\|^2}\right)\right]^+}}\right\}, k \geq 1. \tag{adaPGM}$$

Soon after, adaPGM is generalized to be AdaPG^{q,r} in Latafat et al. [19] with

$$t_k = t_{k-1} \min\left\{\sqrt{\frac{1}{q}+\frac{t_{k-1}}{t_{k-2}}}, \sqrt{\frac{1-r/q}{\left[\frac{t_{k-1}^2\|\nabla f(x^k)-\nabla f(x^{k-1})\|^2+2t_{k-1}(r-1)\left\langle\nabla f(x^k)-\nabla f(x^{k-1}, x^k-x^{k-1}\right\rangle}{\|x^k-x^{k-1}\|^2}-(2r-1)\right]^+}}\right\}, \tag{AdaPG^{q,r}}$$

where $\frac{1}{2} \leq r < p \leq \frac{3+\sqrt{5}}{2}, t_0 = t_{-1} > 0, k \geq 1$. Notably, AdaPG^{q,r} recovers AdPG if $(p,r) = \left(\frac{3}{2}, \frac{3}{4}\right)$ and adaPGM if $(p,r) = \left(1, \frac{1}{2}\right)$ with slight improvements (see [19] for details). The convergence of AdaPG^{q,r} is then established with the worst-case sublinear convergence rate.

1.3 Contributions

In this paper, we will utilize the idea of adaptive stepsize used in Algorithm 1.1 NGD (proposed by Hoai et al.[16]) for proximal gradient scheme (1.3) to solve problem (P) with locally Lipschitz gradient condition imposed on the smooth term f. From now on, we refer to this new method as **NPG** when no confusion arises.

Algorithm 1.1 (NGD) for solving problem $\min\limits_{x\in\mathbb{R}^n} f(x)$ (a special case of Problem (P) when $g=0$) with ∇f being locally Lipschitz.

Step 0 (Initialization). Select $\lambda_0 > 0$, $0 < c_1 < c_0 < \frac{1}{2}$, a tolerance $\varepsilon > 0$ and a positive real sequence $\{\varepsilon_k\}$ such that $\sum\limits_{k=0}^{\infty} \varepsilon_k < \infty$. Choose $x^0 \in \mathbb{R}^n$, $x^1 = x^0 - \lambda_0 \nabla f(x^0), \lambda_{-1} = \lambda_0$ and set $k = 1$.

Step 1.

If $$\|\nabla f(x^k) - \nabla f(x^{k-1})\| > \frac{c_0}{\lambda_{k-1}} \|x^k - x^{k-1}\|$$

then
$$\lambda_k = c_1 \frac{\|x^k - x^{k-1}\|}{\|\nabla f(x^k) - \nabla f(x^{k-1})\|}$$

else $\varepsilon'_{k-1} = \varepsilon_{k-1}$

$$\text{if } \frac{\lambda_{k-1}}{\lambda_{k-2}} < 1 \text{ then } \varepsilon'_{k-1} = \min\left\{\varepsilon_{k-1}, \sqrt{1 + \frac{\lambda_{k-1}}{\lambda_{k-2}}} - 1\right\}$$
$$\lambda_k = (1 + \varepsilon'_{k-1})\lambda_{k-1}.$$

Step 2. Compute $x^{k+1} = x^k - \lambda_k \nabla f(x^k)$.
Step 3. If $\|\nabla f(x^{k+1})\| < \varepsilon$ then STOP
else setting $k := k+1$, and return to **Step 1**.

Particularly, our main contributions are summarized as follows:

1. Firstly, we give a positive response to Question 1.1 by extending NGD to the composite model (P). The resulting algorithm is called **NPG1**. When $g = 0$, **NPG1** becomes NGD but but with an **enhanced** range for the parameters c_0, c_1, extended by a factor of $\sqrt{2}$.

2. To investigate the convergence of **NPG1**, we establish a *sufficient decrease-type inequality*, which ensures the objective function decreases from some fixed iteration. Most importantly, it yields a sublinear convergence rate when f is convex with a locally Lipschitz gradient, and a Q-linear rate when f is locally strongly convex. In contrast, recent methods AdPG [24], adaPGM [20], and AdaPG^{q,r} [19] achieve only *worst-case* sublinear rates. The lack of a descent property in those methods prevents them from obtaining the sublinear rate and Q-linear rate of convergence under assumptions similar to ours.

3. Secondly, **NPG2** is proposed for a broad class of nonconvex f with globally Lipschitz gradient. In this setting, the range of the coefficients c_0, c_1 is extended to $(0,1)$. Importantly, NPG2 achieves the same standard convergence guarantees as classical proximal gradient methods with constant or backtracking stepsizes in the nonconvex setting; see Beck [5, Theorem 10.15].

4. Thirdly, **NPG-quad** is designed for indefinite quadratic functions f with a better approximation for t_k according to the local behavior of f. Moreover, c_0, c_1 are chosen in a larger interval $(0,2)$. The convergence results of **NPG-quad** are shown similarly to those of NPG2.

5. Besides the adaptation in computation of the stepsize selection as the existing methods AdPG, adaPGM, AdaPG^{q,r}, a key distinction of our method **NPG** is that, from a fixed iteration onward, its stepsize sequence is guaranteed to **increase** to a positive limit.

6. Comprehensive experiments are conducted to demonstrate the practical effectiveness of our algorithms on a variety of important problems arising in machine learning, optimal transport, signal processing and related fields.

1.4 Structure of the paper

The rest of the paper is structured as follows. After summarizing some necessary preliminaries in Section 2, we propose our new proximal algorithm in Section 3 for solving the convex situation of (P) under the locally Lipschitz condition of ∇f. In the sequel, we consider a nonconvex case of (P) with an other new algorithm. Section 5 presents a particular version of proposed method applied for the indefinite quadratic function f. The numerical experiments on a set of practical examples are stated in Section 6. Lastly, the paper is closed by some conclusions in Section 7.

2 Preliminaries

In this section, we recall some necessary fundamental results which are useful to derive our main contributions in the upcoming sections.

Definition 2.1 [4] Consider the problem (P) under Assumption 1.

(i) A point x^* at which f is differentiable is called a **stationary point** of (P) if

$$-\nabla f(x^*) \in \partial g(x^*).$$

(ii) For any $t > 0$, the proximal operator $\text{Prox}_{tg} : \mathbb{R}^n \to \mathbb{R}^n$ and the gradient mapping $G^{f,g}_{1/t} : \text{int}(\text{dom}(f)) \to \mathbb{R}^n$ are defined respectively by

$$\text{Prox}_{tg}(y) = \text{argmin}_{x\in\mathbb{R}^n} \left\{ g(x) + \frac{1}{2t}\|x-y\|^2 \right\};$$

$$G^{f,g}_{1/t}(x) = \frac{x - \text{Prox}_{tg}(x - t\nabla f(x))}{t}.$$

When f, g are clear from the context, we shall use the notation $G_{1/t}$ instead of $G^{f,g}_{1/t}$.

Lemma 2.1 (optimality conditions for Problem (P)**, see e.g., Theorem 3.72 in [4])** *Consider the problem* (P) *under Assumption 1.*

(a) If $x^* \in \text{dom}(g)$ is a local optimal solution of (P) and f is differentiable at x^*, then x^* is a stationary point of (P).

(b) Suppose that f is convex. If f is differentiable at $x^* \in \text{dom}(g)$, then x^* is a global optimal solution of (P) if and only x^* is a stationary point of (P).

The following lemmas are derived directly from Theorem 10.7, Theorem 10.9 and Lemma 10.10 in [4].

Lemma 2.2 *Consider the problem* (P) *under Assumption 1 and $t > 0$. Then*

(i) if $g = 0$ then $G_{1/t}(x) = \nabla f(x)$;

(ii) $x^* \in \text{int}(\text{dom}(f))$ is a stationary point of (P) *if and only if $G_{1/t}(x^*) = 0$.*

Lemma 2.3 *Consider the problem* (P) *under Assumption 1, suppose that $0 < t_1 \le t_2$, then*

$$\|G_{1/t_1}(x)\| \ge \|G_{1/t_2}(x)\|, \quad \text{for any } x \in \text{int}(\text{dom}(f)).$$

Lemma 2.4 *Assuming that f, g satisfy Assumption 1 and furthermore ∇f is Lipschitz continuous with constant L_f. Then, for $t > 0$, we have*

$$\|G_{1/t}(x) - G_{1/t}(y)\| \leq \left(\frac{2}{t} + L_f\right)\|x - y\|, \quad \text{for any } x, y \in \text{int}(\text{dom}(f)).$$

Remark 2.1 From the above results, one can use $\|G_{1/t}(x)\|$ as an "optimal measure" for Problem (P) in the sense that "it is always nonnegative and equal to zero if and only if x is a stationary point", see Beck [4] for details.

Lemma 2.5 *Under Assumption 1, the sequence $\{x^k\}$ generated by proximal gradient scheme (1.3) for solving the problem (P) has the following properties:*

(i) there exists $\widetilde{\nabla} g(x^{k+1}) \in \partial g(x^{k+1})$ such that $x^{k+1} = x^k - t_k\left(\nabla f(x^k) + \widetilde{\nabla} g(x^{k+1})\right)$;

(ii) for all $x \in \text{int}(\text{dom}(f))$, we have

$$g(x) - g(x^{k+1}) \geq \left\langle x^{k+1} - x, \nabla f(x^k) + \frac{x^{k+1} - x^k}{t_k}\right\rangle. \tag{2.1}$$

Proof (i) Since $x^{k+1} \in \underset{x\in\mathbb{R}^n}{\text{argmin}}\left\{g(x) + \frac{1}{2t_k}\left\|x - (x^k - t_k\nabla f(x^k))\right\|^2\right\}$ then

$$0 \in \partial g(x^{k+1}) + \frac{1}{t_k}\left(x^{k+1} - x^k + t_k\nabla f(x^k)\right).$$

Hence there exists $\widetilde{\nabla} g(x^{k+1}) \in \partial g(x^{k+1})$ such that

$$x^{k+1} = x^k - t_k(\nabla f(x^k) + \widetilde{\nabla} g(x^{k+1})). \tag{2.2}$$

(ii) From (i) and the convexity of g we easily get that

$$\begin{aligned} g(x) - g(x^{k+1}) &\geq \left\langle x - x^{k+1}, \widetilde{\nabla} g(x^{k+1})\right\rangle \\ &= \left\langle x^{k+1} - x, \nabla f(x^k) + \frac{x^{k+1} - x^k}{t_k}\right\rangle. \end{aligned}$$

The following lemma is a variation of the Opial lemma presented in [23]. This result will be used to analyze the convergence of the proposed method in the next section.

Lemma 2.6 (Lemma 2 [23]) *Let $\{x^k\} \subset \mathbb{R}^n$ be a bounded sequence and its cluster points lie in $X \subset \mathbb{R}^n$. If there exists a nonnegative sequence $\{a_k\} \subset \mathbb{R}_+$ such that*

$$\|x^{k+1} - x\|^2 + a_{k+1} \leq \|x^k - x\|^2 + a_k, \ \ \forall x \in X, \tag{2.3}$$

then $\{x^k\}$ converges to an element of X.

Proof Assume that there exist two different cluster points $\bar{x}^1, \bar{x}^2$ of $\{x^k\}$. Then, there exist two subsequences $x^{k_i} \to \bar{x}^1$ and $x^{k_j} \to \bar{x}^2$. Given that, the real sequence $\|x^k - x\|^2 + a_k$ is lower bounded by zero and nonincreasing, so it converges for any $x \in X$. Let $x = \bar{x}^1$ we have

$$\begin{aligned} \lim_{k\to+\infty}\left(\|x^k - \bar{x}^1\|^2 + a_k\right) &= \lim_{i\to+\infty}\left(\|x^{k_i} - \bar{x}^1\|^2 + a_{k_i}\right) = \lim_{i\to+\infty} a_{k_i} \\ &= \lim_{j\to+\infty}\left(\|x^{k_j} - \bar{x}^1\|^2 + a_{k_j}\right) = \|\bar{x}^2 - \bar{x}^1\|^2 + \lim_{j\to+\infty} a_{k_j}. \end{aligned}$$

Hence, $\lim_{i\to+\infty} a_{k_i} = \lim_{j\to+\infty} a_{k_j} + \|\bar{x}^2 - \bar{x}^1\|^2$. Repeating this with $x = \bar{x}^2$ yields $\lim_{j\to+\infty} a_{k_j} = \lim_{i\to+\infty} a_{k_i} + \|\bar{x}^1 - \bar{x}^2\|^2$. Thus, we obtain $\bar{x}^1 = \bar{x}^2$, which implies the convergence of $\{x^k\}$.

3 A new proximal gradient algorithm for the convex case of the problem (P) with locally Lipschitz ∇f

It is worth noting that when f has a globally Lipschitz gradient with constant L_f and t_k is chosen as a fixed number in $\left(0, \frac{2}{L_f}\right)$ or by some line search strategy, the common technique establishing the convergence of proximal gradient method (1.3) for solving (P) is related to *the sufficient decrease inequality*, i.e., showing the existence of a positive constant M such that

$$F(x^k) - F(x^{k+1}) \geq M\|x^{k+1} - x^k\|, \;\; k \geq 0, \qquad \text{(sufficient decrease ineq.)}$$

For the proximal gradient algorithms using adaptive stepsizes solving problem (P) in the literature like AdPG [24], adaPGM[20] and AdaPG^{q,r} [19], the obstacle of the locally Lipschitz gradient condition has been overcome by constructing Lyapunov type functions and then obtain the boundedness of the iterates. Since, on the compact set $T = \overline{conv}\left(\{x^*, x^0, x^1, ...\}\right)$ ($x^* \in X^*$ is an optimal solution of (P)) all properties of a function f with locally Lipschitz gradient can be operated as those of a globally Lipschitz gradient function. The convergence of their proposed approaches are then deduced by relying on the interesting techniques different from the usual way based on the sufficient decrease ineq. . However, the absence of descent property prevents their algorithms from achieving sublinear convergence rate $O\left(\frac{1}{k}\right)$ of $F(x^k) - F_*$ but only worst-case convergence rate $O\left(\frac{1}{k}\right)$ of $\min\limits_{1\leq i\leq k}\{F(x^i) - F_*\}$ for solving convex problem (P) under locally Lipschitz gradient condition. Notably, our stepsize selection **NPG** is not only adapted with the local curvature of f but also controllable by using the pre-selected positive convergent series $\sum\limits_{k=0}^{+\infty}\gamma_k$. Then, in contrast of the existing algorithms, our proposed algorithms based on **NPG** stepsize can establish sufficient decrease ineq. for (P). We will successively explore how to establish this inequality in the subsequent parts of the paper.

Firstly, in this section, we propose a new proximal gradient algorithm **NPG1** to solve problem (P) under *Assumption 1* and *Assumption 2* below.

Assumption 2 *f is convex and has a locally Lipschitz gradient.*

Algorithm 3.1 (NPG1)

Step 0. Select $t_0 > 0$, $0 < c_1 < c_0 < \frac{1}{\sqrt{2}}$, a tolerance $\varepsilon > 0$ and a positive real sequence $\{\gamma_k\}$ such that $\sum\limits_{k=0}^{+\infty}\gamma_k < \infty$. Choose $x^0 \in \text{int}(\text{dom}(f))$, $x^1 = \text{Prox}_{t_0 g}(x^0 - t_0\nabla f(x^0)), t_{-1} = t_0$ and set $k = 1$.
Step 1.

$$\textbf{If} \quad \|\nabla f(x^k) - \nabla f(x^{k-1})\| > \frac{c_0}{t_{k-1}}\|x^k - x^{k-1}\| \tag{3.1}$$

$$\textbf{then} \quad t_k = c_1\frac{\|x^k - x^{k-1}\|}{\|\nabla f(x^k) - \nabla f(x^{k-1})\|} \tag{3.2}$$

$$\textbf{else} \quad \gamma'_{k-1} = \gamma_{k-1}$$

$$\text{if } \frac{t_{k-1}}{t_{k-2}} < 1 \text{ then } \gamma'_{k-1} = \min\left\{\gamma_{k-1}, \sqrt{1 + \frac{t_{k-1}}{t_{k-2}}} - 1\right\} \tag{3.3}$$

$$t_k = (1 + \gamma'_{k-1})t_{k-1}. \tag{3.4}$$

Step 2. Compute $x^{k+1} = \text{Prox}_{t_k g}(x^k - t_k\nabla f(x^k))$.
Step 3. If $\|G_{1/t_k}(x^k)\| = \|x^k - x^{k+1}\|/t_k < \varepsilon$ **then** STOP **else** setting $k := k + 1$ and return to **Step 1**.

Remark 3.1 It is observed that, in the case $g = 0$, Algorithm 3.1 (NPG1) becomes NGD [16] with larger bounds of c_0, c_1. In particular, for NGD, $c_0, c_1 \in \left(0, \frac{1}{2}\right)$ but for NPG1, $c_0, c_1 \in \left(0, \frac{1}{\sqrt{2}}\right)$. The key

improvement lies in how we estimate the term $t_k^2\left\|\nabla f(x^k)+\widetilde{\nabla} g(x^k)\right\|^2$ in Lemma 3.3(i). Specifically, when $g=0$, this quantity in NPG1 reduces to $\|x^{k+1}-x^k\|^2$ in NGD [16], where its upper bound was derived using the Cauchy–Schwarz inequality. In contrast, our approach avoids that technique and instead relies solely on the convexity of f and g. As a result, the upper bounds for c_0 and c_1 are improved over those in NGD.

Analogous to existing methods we need to prepare some lemmas which will help us to prove the boundedness of $\{x^k\}$ - a key step to overcome the difficulties generated by the locally Lipschitz continuity of ∇f.

Lemma 3.1 *For all $x\in \mathrm{int}(\mathrm{dom}(f))$ we have*

$$\|x^{k+1}-x\|^2+2t_k\left(F(x^k)-F(x)\right)\leq \|x^k-x\|^2+t_k^2\left\|\nabla f(x^k)+\widetilde{\nabla} g(x^k)\right\|^2.$$

Proof From Lemma 2.5 (ii), for all $x\in \mathrm{int}(\mathrm{dom}(f))$

$$\begin{aligned}2t_k\left(g(x^{k+1})-g(x)\right)&\leq 2\left\langle x^{k+1}-x^k+t_k\nabla f(x^k),x-x^{k+1}\right\rangle\\&=\|x^k-x\|^2-\|x^{k+1}-x^k\|^2-\|x^{k+1}-x\|^2+\\&\quad+2t_k\left\langle\nabla f(x^k),x-x^{k+1}\right\rangle.\end{aligned}\tag{3.5}$$

Using the convexity of f and g, we continue evaluating

$$\begin{aligned}\langle\nabla f(x^k),x-x^{k+1}\rangle&=\langle\nabla f(x^k),x-x^k\rangle+\langle\nabla f(x^k)+\widetilde{\nabla} g(x^k),x^k-x^{k+1}\rangle+\langle\widetilde{\nabla} g(x^k),x^{k+1}-x^k\rangle\\&\leq f(x)-f(x^k)+\left\langle\nabla f(x^k)+\widetilde{\nabla} g(x^k),x^k-x^{k+1}\right\rangle+g(x^{k+1})-g(x^k).\end{aligned}\tag{3.6}$$

From (3.5) and (3.6), we derive that

$$\|x^{k+1}-x\|^2+2t_k\left(F(x^k)-F(x)\right)\leq\|x^k-x\|^2+R,\tag{3.7}$$

where

$$\begin{aligned}R&=2t_k\left\langle\nabla f(x^k)+\widetilde{\nabla} g(x^k),x^k-x^{k+1}\right\rangle-\|x^{k+1}-x^k\|^2\\&=t_k\left\langle 2\nabla f(x^k)+2\widetilde{\nabla} g(x^k)-\nabla f(x^k)-\widetilde{\nabla} g(x^{k+1}),x^k-x^{k+1}\right\rangle\\&=t_k^2\left\langle\nabla f(x^k)+2\widetilde{\nabla} g(x^k)-\widetilde{\nabla} g(x^{k+1}),\nabla f(x^k)+\widetilde{\nabla} g(x^{k+1})\right\rangle\\&=t_k^2\left(\left\|\nabla f(x^k)+\widetilde{\nabla} g(x^k)\right\|^2-\left\|\widetilde{\nabla} g(x^{k+1})-\widetilde{\nabla} g(x^k)\right\|^2\right)\\&\leq t_k^2\left\|\nabla f(x^k)+\widetilde{\nabla} g(x^k)\right\|^2.\end{aligned}\tag{3.8}$$

The final conclusion is obtained by (3.7) and (3.8).

Lemma 3.2 *Let $\{t_k\}$ be a sequence of stepsizes generated by Algorithm 3.1 then there exists $k_0\in\mathbb{N}$ such that*

$$1+\frac{t_k}{t_{k-1}}\geq\frac{t_{k+1}^2}{t_k^2}\ \ \forall k\geq k_0.\tag{3.9}$$

Proof If $\|\nabla f(x^{k+1})-\nabla f(x^k)\|>\frac{c_0}{t_k}\|x^{k+1}-x^k\|$ then $t_{k+1}=\frac{c_1\|x^{k+1}-x^k\|}{\|\nabla f(x^{k+1})-\nabla f(x^k)\|}<\frac{c_1t_k}{c_0}$ (by (3.2)). Hence $\frac{t_{k+1}}{t_k}<\frac{c_1}{c_0}<1$ and (3.9) is followed. Conversely, in the case that $\|\nabla f(x^{k+1})-\nabla f(x^k)\|\leq\frac{c_0}{t_k}\|x^{k+1}-x^k\|$ then by (3.4), $t_{k+1}=(1+\gamma_k')t_k$ and (3.9) is equivalent to

$$\left(\frac{t_{k+1}}{t_k}\right)^2=(1+\gamma_k')^2\leq 1+\frac{t_k}{t_{k-1}}.\tag{3.10}$$

Moreover, from (3.3), if $\frac{t_k}{t_{k-1}} \geq 1$ then $\gamma'_k = \gamma_k$, furthermore since $\sum_{k=0}^{+\infty} \gamma_k < +\infty$,, we have $\gamma_k \to 0$, as $k \to +\infty$, hence there exists k_0 such that

$$\gamma'_k = \gamma_k \leq \sqrt{2} - 1 \leq \sqrt{1 + \frac{t_k}{t_{k-1}}} - 1 \quad \forall k \geq k_0. \tag{3.11}$$

For the remaining case $\frac{t_k}{t_{k-1}} < 1$, we have

$$\gamma'_k = \min\left\{\gamma_k, \sqrt{1 + \frac{t_k}{t_{k-1}}} - 1\right\} \leq \sqrt{1 + \frac{t_k}{t_{k-1}}} - 1. \tag{3.12}$$

Thus, (3.9) is proved from (3.11) and (3.12).

As mentioned above, the bounded property of the sequence $\{x^k\}$ in the following lemma provides us an important key beyond the challenge of locally Lipschitz continuity of ∇f.

Lemma 3.3 *Let $\{x^k\}$ be a sequence generated by Algorithm 3.1 then the following statements hold*

(i) there exists $k_1 \geq k_0$ such that for all $k \geq k_1$,

$$t_k^2 \left\|\nabla f(x^k) + \widetilde{\nabla} g(x^k)\right\|^2 \leq \frac{1}{2}\|x^k - x^{k-1}\|^2 + \frac{t_k^2}{t_{k-1}}\left(F(x^{k-1}) - F(x^k)\right); \tag{3.13}$$

(ii) $\{x^k\}$ is bounded.

Proof (i) We have the relation

$$t_k^2 \left\|\nabla f(x^k) + \widetilde{\nabla} g(x^k)\right\|^2 = \underbrace{t_k^2 \left\|\nabla f(x^k) - \nabla f(x^{k-1})\right\|^2}_{A} + B, \tag{3.14}$$

where

$$\begin{aligned} B &= 2t_k^2 \left\langle \nabla f(x^k) + \widetilde{\nabla} g(x^k), \nabla f(x^{k-1}) + \widetilde{\nabla} g(x^k)\right\rangle - t_k^2 \left\|\nabla f(x^{k-1}) + \widetilde{\nabla} g(x^k)\right\|^2 \\ &= \frac{t_k^2}{t_{k-1}} \left\langle \nabla f(x^k) + \widetilde{\nabla} g(x^k), x^{k-1} - x^k\right\rangle + \frac{t_k^2}{t_{k-1}} \underbrace{\left\langle \nabla f(x^k) - \nabla f(x^{k-1}), x^{k-1} - x^k\right\rangle}_{\leq 0} \\ &\leq \frac{t_k^2}{t_{k-1}}\left(F(x^{k-1}) - F(x^k)\right). \end{aligned} \tag{3.15}$$

We now prove that there exists $k_1 \geq k_0$ such that

$$A \leq \frac{1}{2}\|x^k - x^{k-1}\|^2 \quad \forall k \geq k_1. \tag{3.16}$$

Indeed, from Algorithm 3.1, if $\left\|\nabla f(x^k) - \nabla f(x^{k-1})\right\| > \frac{c_0}{t_{k-1}}\|x^k - x^{k-1}\|$ then $t_k = \frac{c_1\|x^k - x^{k-1}\|}{\|\nabla f(x^k) - \nabla f(x^{k-1})\|}$ and since $c_1 < \frac{1}{\sqrt{2}}$, we have

$$A = t_k^2\|\nabla f(x^k) - \nabla f(x^{k-1})\|^2 = c_1^2\|x^k - x^{k-1}\|^2 < \frac{1}{2}\|x^k - x^{k-1}\|^2.$$

Conversely, if $\|\nabla f(x^k) - \nabla f(x^{k-1})\| \leq \frac{c_0}{t_{k-1}}\|x^k - x^{k-1}\|$ then

$$t_k = (1 + \gamma'_{k-1})t_{k-1} \leq (1 + \gamma_{k-1})\frac{c_0\|x^k - x^{k-1}\|}{\|\nabla f(x^k) - \nabla f(x^{k-1})\|}$$

which follows

$$t_k^2\|\nabla f(x^k)-\nabla f(x^{k-1})\|^2 \le (1+\gamma_{k-1})^2 c_0^2\|x^k-x^{k-1}\|^2. \tag{3.17}$$

The convergence of $\sum\limits_{k=0}^{+\infty}\gamma_k$ indicates that there exists $k_1 \ge k_0$ satisfying

$$\gamma_{k-1} \le \frac{1}{\sqrt{2}c_0}-1 \quad \forall k \ge k_1 \ \left(\frac{1}{\sqrt{2}c_0}-1>0 \text{ since } c_0<\frac{1}{\sqrt{2}}\right), \tag{3.18}$$

which is equivalent to $(1+\gamma_{k-1})^2c_0^2 \le \frac{1}{2}$ for all $k \ge k_1$. From (3.17) we have (3.16). The combination of (3.14), (3.15) and (3.16) indicates (3.13).

(ii) Using Lemma 3.1 with $x = x^*$ and (3.13), for all $k \ge k_1$ we have

$$\begin{aligned}
&\|x^{k+1}-x^*\|^2 + 2t_k\left(F(x^k)-F(x^*)\right) + t_k^2\left\|\nabla f(x^k)+\widetilde{\nabla} g(x^k)\right\|^2\\
&\le \|x^k-x^*\|^2 + 2t_k^2\left\|\nabla f(x^k)+\widetilde{\nabla} g(x^k)\right\|^2\\
&\le \|x^k-x^*\|^2 + \|x^k-x^{k-1}\|^2 + 2\frac{t_k^2}{t_{k-1}}\left(F(x^{k-1})-F(x^k)\right).
\end{aligned} \tag{3.19}$$

Nevertheless,

$$\begin{aligned}
t_k^2\left\|\nabla f(x^k)+\widetilde{\nabla} g(x^k)\right\|^2 &= \left\|t_k\left(\nabla f(x^k)+\widetilde{\nabla} g(x^{k+1})\right)+t_k\left(\widetilde{\nabla} g(x^k)-\widetilde{\nabla} g(x^{k+1})\right)\right\|^2\\
&= \left\|(x^k-x^{k+1})+t_k\left(\widetilde{\nabla} g(x^k)-\widetilde{\nabla} g(x^{k+1})\right)\right\|^2\\
&= \|x^k-x^{k+1}\|^2 + \underbrace{2t_k\left\langle x^k-x^{k+1}, \widetilde{\nabla} g(x^k)-\widetilde{\nabla} g(x^{k+1})\right\rangle}_{\ge 0 \text{ because } g \text{ is convex}} + t_k^2\underbrace{\|\widetilde{\nabla} g(x^k)-\widetilde{\nabla} g(x^{k+1})\|^2}_{\ge 0}\\
&\ge \|x^k-x^{k+1}\|^2.
\end{aligned} \tag{3.20}$$

Hence, using inequality (3.20) for the left hand side of (3.19) we obtain that

$$\begin{aligned}
&\|x^{k+1}-x^*\|^2 + 2t_k\left(1+\frac{t_k}{t_{k-1}}\right)\left(F(x^k)-F(x^*)\right) + \|x^k-x^{k+1}\|^2\\
&\le \|x^k-x^*\|^2 + \|x^{k-1}-x^k\|^2 + 2\frac{t_k^2}{t_{k-1}}\left(F(x^{k-1})-F(x^*)\right).
\end{aligned} \tag{3.21}$$

Remember that from Lemma 3.2 we derive $2t_k\left(1+\frac{t_k}{t_{k-1}}\right) \ge \frac{2t_{k+1}^2}{t_k}\ \forall\, k \ge k_1$. Therefore, by (3.21), for all $k \ge k_1$ we have

$$\begin{aligned}
&\|x^{k+1}-x^*\|^2 + \|x^k-x^{k+1}\|^2 + \frac{2t_{k+1}^2}{t_k}\left(F(x^k)-F(x^*)\right)\\
\le\ &\|x^k-x^*\|^2 + \|x^{k-1}-x^k\|^2 + \frac{2t_k^2}{t_{k-1}}\left(F(x^{k-1})-F(x^*)\right).
\end{aligned} \tag{3.22}$$

This inequality follows that

$$\|x^{k+1}-x^*\|^2 + \|x^k-x^{k+1}\|^2 + \frac{2t_{k+1}^2}{t_k}\left(F(x^k)-F(x^*)\right) \le K, \ \ \forall k \ge k_1, \tag{3.23}$$

where

$$K = \|x^{k_1}-x^*\|^2 + \|x^{k_1-1}-x^{k_1}\|^2 + \frac{2t_{k_1}^2}{t_{k_1-1}}\left(F(x^{k_1-1})-F(x^*)\right).$$

The relation (3.23) implies the boundedness of $\{x^k\}$.

Remark 3.2 From the proof of Lemma 3.3 (eq. (3.11) and (3.18)), we see that if the convergent positive series $\sum_{k=0}^{+\infty}\gamma_k$ is created such that $\gamma_k \le \min\left\{\frac{1}{\sqrt{2}c_0}-1, \sqrt{2}-1\right\}$ for all $k \ge 1$ then $k_1 = 1$ and therefore we obtain (3.22) for all $k \ge 1$. Consequently, by using arguments analogous to those given by Malitsky and Mishchenko [24] we immediately obtain all similar convergent results of NPG1 as that of AdPG [24] such as the *worst-case* sublinear convergence of $\{x^k\}$ to an optimal solution of problem (P).

Nevertheless, one will see in the upcoming parts of the paper, we analyze the convergent results of NPG1 by designing a sufficient decrease inequality (in Corollary 3.1) without globally Lipschitz assumption on $\nabla f(x)$. This technique is different from that of [24,20,19]. To get this, in the sequel, we deploy the special properties of $\{t_k\}$ presented in the following lemma.

Lemma 3.4 *Let $\{t_k\}$ be a sequence of stepsizes generated by Algorithm 3.1. Then,*

(i) $\{t_k\}$ is lower bounded by a positive number;

(ii) $\{t_k\}$ is convergent and has a positive limit.

Proof (i) By Lemma 3.3 the set $T = \overline{conv}\left(\{x^*, x^0, x^1, ...\}\right)$ is bounded and closed hence it is compact. From the local Lipschitz continuity of ∇f, it is easy to see that there exists $L_0 > 0$ satisfying $\|\nabla f(x) - \nabla f(y)\| \le L_0\|x-y\| \quad \forall x, y \in T$. Thereafter, either $t_1 \ge \frac{c_1}{L_0}$ or $t_1 = (1+\gamma_0')t_0 \ge t_0$. The induction process derives that

$$t_k \ge \min\left\{\frac{c_1}{L_0}, t_0\right\} = t_{\min} > 0 \quad \forall k \ge 0. \tag{3.24}$$

(ii) If we set $r_k = \ln t_{k+1} - \ln t_k$ and $r_k^+ = \max\{0, r_k\} \ge 0, r_k^- = -\min\{0, r_k\} \ge 0, \forall k \ge 0$ then $r_k = r_k^+ - r_k^-$. On the other hand, from Algorithm 3.1, we observe that $0 < c_1 < c_0 < \frac{1}{\sqrt{2}}$, hence both of (3.2) and (3.4) give

$$r_k = \ln\frac{t_{k+1}}{t_k} \le \ln(1+\gamma_k') \le \gamma_k' \le \gamma_k \quad \forall k \ge 0.$$

Thus, $r_k^+ \le \gamma_k$. Moreover, the series $\sum_{k=0}^{+\infty}\gamma_k$ converges then $\sum_{k=0}^{+\infty} r_k^+ < +\infty$. Noticeably,

$$\ln t_{k+1} - \ln t_0 = \sum_{i=0}^{k} r_i = \sum_{i=0}^{k}(r_i^+ - r_i^-) = \sum_{i=0}^{k} r_i^+ - \sum_{i=0}^{k} r_i^-. \tag{3.25}$$

Hence if the nonnegative series $\sum_{k=0}^{+\infty} r_k^-$ diverges, i.e., $\lim_{k\to+\infty}\sum_{i=0}^{k} r_i^- = +\infty$ then

$$\lim_{k\to+\infty}(\ln t_{k+1}) = -\infty$$

which implies $\lim_{k\to+\infty} t_k = 0$. This result is contradict with the assertion (i). Thus, $\sum_{k=0}^{+\infty} r_k^-$ is convergent and therefore $\lim_{k\to+\infty} t_k = t^* \in (0, +\infty)$ (followed by (3.25)).

The result in the following lemma gives us an inequality like Lipschitz gradient continuity but with flexible constant for each pair of x^{k-1} and x^k.

Lemma 3.5 *There exists k^* such that*

$$\|\nabla f(x^k) - \nabla f(x^{k-1})\| \le \frac{c_0}{t_{k-1}}\|x^k - x^{k-1}\|, \ \ \forall k \ge k^*. \tag{3.26}$$

Proof Assuming that there is a subsequence $\{k_i\} \subset \mathbb{N}$, $k_i \to +\infty$ such that

$$\|\nabla f(x^{k_i}) - \nabla f(x^{k_i-1})\| > \frac{c_0}{t_{k_i-1}}\|x^{k_i} - x^{k_i-1}\|.$$

By Algorithm 3.1, in this case we have

$$\frac{t_{k_i}}{t_{k_i-1}} = \frac{c_1\|x^{k_i}-x^{k_i-1}\|}{t_{k_i-1}\|\nabla f(x^{k_i})-\nabla f(x^{k_i-1})\|} < \frac{c_1}{c_0} \quad \forall k_i.$$

However, Lemma 3.4 gives

$$\lim_{k_i\to+\infty} t_{k_i} = \lim_{k_i\to+\infty} t_{k_i-1} = \lim_{k\to+\infty} t_k = t^*.$$

Consequently, $\frac{t^*}{t^*} \le \frac{c_1}{c_0} < 1$ that is impossible and we obtain the conclusion of the lemma.

Remark 3.3 From Lemma 3.5, we immediately obtain the increasing of the sequence $\{t_k\}_{k\ge k^*}$, i.e., $t_{k^*} \le t_k \le t^*$ for all $k \ge k^*$ and therefore $0 < t_{\min} \le t_k \le \max\{t_0,...,t_{k^*-1},t^*\} = t_{max},\ k \ge 0$.

The next lemma plays a crucial role in proving the convergence of Algorithm 3.1 (NPG1).

Lemma 3.6 *For any $x \in \text{int}(\text{dom}(f))$, we have*

$$F(x) - F(x^{k+1}) \ge \frac{1-c_0}{t_k}\|x^{k+1}-x^k\|^2 + \frac{1}{t_k}\langle x^k - x^{k+1}, x - x^k\rangle, \quad \textit{for all } k \ge k^*. \tag{3.27}$$

Proof Because of the convexity of f and Lemma 2.5 (ii) we have

$$\begin{aligned}
&F(x) - F(x^{k+1}) = f(x) + g(x) - f(x^{k+1}) - g(x^{k+1}) \\
&\ge f(x^k) + \left\langle x - x^k, \nabla f(x^k)\right\rangle + \left\langle x^{k+1} - x, \nabla f(x^k) + \frac{x^{k+1}-x^k}{t_k}\right\rangle - f(x^{k+1}) \\
&= f(x^k) - f(x^{k+1}) + \left\langle x^{k+1} - x^k, \nabla f(x^k)\right\rangle + \frac{1}{t_k}\left\langle x^{k+1}-x^k, x^{k+1}-x\right\rangle \\
&\ge \langle \nabla f(x^{k+1}) - \nabla f(x^k), x^k - x^{k+1}\rangle + \frac{1}{t_k}\left\|x^{k+1}-x^k\right\|^2 + \frac{1}{t_k}\left\langle x^{k+1}-x^k, x^k - x\right\rangle.
\end{aligned} \tag{3.28}$$

On the other hand, by using Lemma 3.5, we have the evaluation

$$\begin{aligned}
\left\langle \nabla f(x^{k+1}) - \nabla f(x^k), x^k - x^{k+1}\right\rangle &\ge -\left\|\nabla f(x^k) - \nabla f(x^{k+1})\right\|\|x^k - x^{k+1}\| \\
&\ge -\frac{c_0}{t_k}\|x^{k+1}-x^k\|^2 \quad \forall k \ge k^*.
\end{aligned} \tag{3.29}$$

The proof is completed by utilizing (3.28) and (3.29).

It is observed that if we substitute x by x^k in (3.27) of Lemma 3.6 and using Remark 3.3 we immediately get the following corollary known as a sufficient decrease type inequality.

Corollary 3.1 (sufficient decrease type inequality) *For all $k \ge k^*$ we have*

$$F(x^k) - F(x^{k+1}) \ge \frac{1-c_0}{t_k}\|x^{k+1}-x^k\|^2.$$

Combining with the fact that $t_{k^*} \le t_k \le t^*$ for all $k \ge k^*$ we get the following inequalities

$$F(x^k) - F(x^{k+1}) \ge \frac{1-c_0}{t^*}\|x^{k+1}-x^k\|^2 \ge 0 \tag{3.30}$$

and for all $k \ge k^*$,

$$F(x^k) - F(x^{k+1}) \ge \frac{1-c_0}{t_k}\|x^{k+1}-x^k\|^2 = (1-c_0)t_k\|G_{1/t_k}(x^k)\|^2 \ge (1-c_0)t_{k^*}\|G_{1/t_k}(x^k)\|^2. \tag{3.31}$$

Now we are ready to establish the convergent properties of Algorithm 3.1 (NPG1) in the following theorem.

Theorem 3.1 (the convergence of NPG1) *Suppose that problem (P) satisfies Assumptions 1 and 2. Then the following assertions hold for Algorithm 3.1.*

(i) The sequence $\{F(x^k)\}_{k\geq k^*}$ *descends to* $\lim_{k\to+\infty} F(x^k) = F_*$.

(ii) The sequence $\{x^k\}$ *converges to an optimal solution of problem (P).*

(iii) For any $x^* \in X^*$ *and* $k \geq k^* + 1$ *we have*

$$F(x^k) - F_* = F(x^k) - F(x^*) \leq \frac{D}{2t_{k^*}(k-k^*)} = O\left(\frac{1}{k}\right), \tag{3.32}$$

where

$$D = \max\left\{\|x^* - x^{k^*}\|^2, \|x^* - x^{k^*}\|^2 + \frac{t^*(2c_0-1)}{1-c_0}\left(F(x^{k^*}) - F_*\right)\right\}.$$

Proof (i) By (3.30), the sequence $\{F(x^k)\}_{k\geq k^*}$ is decreasing. On the other hand, it is lower bounded by F_* hence converges to $\overline{F} \geq F_*$. Thus, $F(x^k) - F(x^{k+1}) \to 0$. And consequently, the inequality (3.30) follows

$$\lim_{k\to+\infty} \|x^{k+1} - x^k\| = 0. \tag{3.33}$$

Now, replacing x with x^* in (3.27) of Lemma 3.6 to obtain

$$\begin{aligned} 0 \leq F(x^{k+1}) - F(x^*) &\leq -\frac{1-c_0}{t_k}\|x^{k+1} - x^k\|^2 - \frac{1}{t_k}\langle x^k - x^{k+1}, x^* - x^k\rangle \\ &\leq \frac{(c_0-1)\|x^{k+1} - x^k\|^2 + \|x^{k+1} - x^k\|\|x^k - x^*\|}{t_k}, \quad \text{for all } k \geq k^*. \end{aligned} \tag{3.34}$$

However, $\{x^k\}$ is bounded (by Lemma 3.3(ii)) and $\lim_{k\to+\infty} t_k = t^*$ (from Lemma 3.4) then combining with (3.33) we deduce that the limit of the right hand side of (3.34) is zero as k tending to infinity. Hence, again, by (3.34) we have $\lim_{k\to+\infty} F(x^k) = F_*$.

(ii) Taking into account that the sequence $\{x^k\}$ is bounded then for each cluster point $\bar{x}$ of $\{x^k\}$, we can take a subsequence $\{x^{k_i}\}$ such that $x^{k_i} \to \bar{x}$. On the other hand, the closedness of F (from *Assumption 1*) follows its lower semi-continuity and therefore $F(\bar{x}) \leq \lim_{k_i\to\infty} F(x^{k_i}) = F_*$, which implies $\bar{x} \in X^*$.
Setting $a_k = \|x^{k-1} - x^k\|^2 + \frac{2t_k^2}{t_{k-1}}\left(F(x^{k-1}) - F(x^*)\right) \geq 0$ and rewrite (3.22) to be

$$\|x^{k+1} - x^*\|^2 + a_{k+1} \leq \|x^k - x^*\|^2 + a_k, \ \ \forall x^* \in X^*, \ \ k \geq k_1.$$

Moreover, we have just shown that all cluster points of $\{x^k\}$ belong to X^*. Therefore, applying Lemma 2.6 we obtain that $\{x^k\}$ converges to some element of X^*.

(iii) In (3.30), substituting k by j then summing up it from $j = k^*$ to k we derive that

$$F(x^{k^*}) - F(x^{k+1}) \geq \frac{1-c_0}{t^*}\sum_{j=k^*}^{k}\|x^{j+1} - x^j\|^2. \tag{3.35}$$

This indicates the convergence of $\sum_{j=k^*}^{+\infty}\|x^{j+1} - x^j\|^2$ and

$$\sum_{j=k^*}^{+\infty}\|x^{j+1} - x^j\|^2 \leq \frac{t^*}{1-c_0}\left(F(x^{k^*}) - F_*\right). \tag{3.36}$$

Applying (3.27) again, we obtain that

$$F(x^*) - F(x^{j+1}) \geq \frac{1}{2t_j}\left(\|x^{j+1} - x^j\|^2 + 2\left\langle x^j - x^{j+1}, x^* - x^j\right\rangle\right) + \left(\frac{1}{2} - c_0\right)\frac{\|x^j - x^{j+1}\|^2}{t_j}$$
$$\geq \frac{1}{2t_j}\left(\|x^* - x^{j+1}\|^2 - \|x^* - x^j\|^2\right) + \left(\frac{1}{2} - c_0\right)\frac{\|x^j - x^{j+1}\|^2}{t_j} \quad \forall j \geq k^*. \tag{3.37}$$

On the other hand, Remark 3.3 gives $t_j \geq t_{k^*}\ \forall j \geq k^*$ which helps to infer the following inequality from (3.37)

$$2t_{k^*}\left(F(x^{j+1}) - F(x^*)\right) \leq 2t_j\left(F(x^{j+1}) - F(x^*)\right)$$
$$\leq \left(\|x^* - x^j\|^2 - \|x^* - x^{j+1}\|^2\right) + (2c_0 - 1)\|x^j - x^{j+1}\|^2 \quad \forall j \geq k^*. \tag{3.38}$$

Summing (3.38) side by side for $j = k^*$ to $k + k^* - 1$ $(k \geq 1)$, we get that

$$2t_{k^*}\left(\sum_{j=k^*}^{k+k^*-1} F(x^{j+1}) - kF(x^*)\right) \leq \left(\|x^* - x^{k^*}\|^2 - \|x^* - x^{k+k^*}\|^2\right) +$$
$$+ (2c_0 - 1)\sum_{j=k^*}^{k+k^*-1}\|x^j - x^{j+1}\|^2$$
$$\leq \|x^* - x^{k^*}\|^2 + (2c_0 - 1)\sum_{j=k^*}^{k+k^*-1}\|x^j - x^{j+1}\|^2 \tag{3.39}$$
$$\leq D, \tag{3.40}$$

where, (from (3.36))D is defined by

$$D = \max\left\{\|x^* - x^{k^*}\|^2, \|x^* - x^{k^*}\|^2 + \frac{t^*(2c_0 - 1)}{1 - c_0}\left(F(x^{k^*}) - F_*\right)\right\}.$$

Additionally, the descent of $\{F(x^k)\}_{k \geq k^*}$ induces $\sum_{j=k^*}^{k+k^*-1} F(x^{j+1}) \geq kF(x^{k+k^*})$. Therefore, by (3.40), we have

$$F(x^{k+k^*}) - F(x^*) \leq \frac{1}{2t_{k^*}}\frac{D}{k} \quad \forall k \geq 1,$$

which means that

$$F(x^k) - F(x^*) \leq \frac{D}{2t_{k^*}}\frac{1}{k - k^*} = O\left(\frac{1}{k}\right) \quad \forall k \geq k^* + 1. \tag{3.41}$$

Next, we discuss the complexity bound of Algorithm 3.1 with respect to tolerance ε in the following remark.

Remark 3.4 From (3.31) we derive that for all $k \geq k^*$,

$$F(x^k) - F_* \geq F(x^{k+1}) - F_* + (1 - c_0)t_{k^*}\|G_{1/t_k}(x^k)\|^2, \tag{3.42}$$

Taking an integer number $q \geq k^*$ then summing (3.42) from $k = q$ to $2q - 1$ yields that

$$F(x^q) - F_* \geq F(x^{2q}) - F_* + (1 - c_0)t_{k^*}\sum_{k=q}^{2q-1}\|G_{1/t_k}(x^k)\|^2. \tag{3.43}$$

Now, remember that $F(x^{2q}) - F_* \geq 0$ and combining (3.43) with (3.41) we obtain

$$(1 - c_0)t_{k^*}\sum_{k=q}^{2q-1}\|G_{1/t_k}(x^k)\|^2 \leq \frac{D}{2t_{k^*}}\frac{1}{q - k^*}. \tag{3.44}$$

Hence

$$\min_{k=q,\dots,2q-1} \|G_{1/t_k}(x^k)\|^2 \leq \frac{D}{2(1-c_0)t_{k^*}^2}\frac{1}{q(q-k^*)}. \tag{3.45}$$

Therefore, we have for any $K \geq 2k^*$,

$$\min_{k=0,\dots,K} \|G_{1/t_k}(x^k)\| \leq \sqrt{\frac{2D}{(1-c_0)t_{k^*}^2}\frac{1}{K(K-2k^*)}} = O\left(\frac{1}{K}\right). \tag{3.46}$$

The final result of this section establishes a stronger convergence rate of Algorithm 3.1 if f is locally strongly convex. The detail is as follows.

Theorem 3.2 *Assuming that $c_0 \leq \frac{1}{2}$ and f is locally strongly convex then under Assumptions 1, 2, the sequence $\{x^k\}$ generated by Algorithm 3.1 satisfies*

$$\|x^{k+1}-x^*\|^2 \leq (1-\sigma t_{k^*})\|x^k-x^*\|^2, \ \ \forall k \geq k^*, \tag{3.47}$$

where, $x^* \in X^*$ and $\sigma > 0$ is strong convexity constant of f on the compact set $T = \overline{conv}\left(\{x^*, x^0, x^1, \dots\}\right)$. Consequently, this result shows the Q-linear convergence rate of $\{x^k\}$.

Proof The $\sigma-$ strong convexity on T of f implies that

$$f(x) - f(x^k) \geq \langle \nabla f(x^k), x-x^k\rangle + \frac{\sigma}{2}\|x-x^k\|^2, \ \forall x \in T.$$

We update this change and the condition $c_0 \leq \frac{1}{2}$ in the argument of formula (3.28) and (3.37) to obtain the following inequality

$$F(x^*) - F(x^{k+1}) \geq \frac{1}{2t_k}\|x^*-x^{k+1}\|^2 + \left(\frac{\sigma}{2} - \frac{1}{2t_k}\right)\|x^*-x^k\|^2,$$

for $x^* \in X^*, k \geq k^*$, Remember that $F(x^*) - F(x^{k+1}) \leq 0 \ \forall k$ hence

$$\frac{1}{2t_k}\|x^*-x^{k+1}\|^2 \leq \left(\frac{1}{2t_k} - \frac{\sigma}{2}\right)\|x^*-x^k\|^2, \ \ k \geq k^*. \tag{3.48}$$

By (3.48), Lemma 3.4(i) and Remark 3.3, we have: $\forall k \geq k^*$

$$0 < 1-\sigma t_k \leq 1-\sigma t_{k^*} \leq 1-\sigma t_{\min} < 1,$$

which derives

$$\|x^{k+1}-x^*\|^2 \leq (1-\sigma t_{k^*})\|x^k-x^*\|^2, \ \ k \geq k^*.$$

The last inequality demonstrates the Q-linear convergence rate of $\{x^k\}$.

4 For a class of the nonconvex case of problem (P)

We now consider Problem (P) satisfying *Assumption 1* and other conditions in *Assumption 3* below

Assumption 3 *(i) f has a globally Lipschitz gradient with constant L_f on* $\text{int}(\text{dom}(f))$.

(ii) For $u, v \in \text{int}(\text{dom}(f))$, the function $h_{uv} : [0,1] \to \mathbb{R}$ defined by

$$h_{uv}(t) = f_t'(u+t(v-u)) = \langle \nabla f(u+t(v-u)), v-u\rangle$$

is quasiconvex.

Example 4.1 Suppose that f is either convex or concave. Then f satisfies *Assumption 3 (ii)*. Indeed, the convexity (concavity, resp.) of f follows the convexity (concavity, resp.) of $f(u+t(v-u))$ on the set $\{t \in \mathbb{R} \mid u+t(v-u) \in \text{int}(\text{dom}(f))\} \supset [0,1]$ (since $\text{int}(\text{dom}(f))$ is convex). As a result, $f_t'(u+t(v-u))$ is increasing (decreasing, resp.) monotone over $[0,1]$ and therefore quasiconvex on that. In the case where f is a concave function then $F = f+g$ is actually the difference of the two convex functions, or in other words, F belongs to the class of *DC functions*.

Example 4.2 The indefinite quadratic function $f(x) = \frac{1}{2}x^TAx+b^Tx$ (A is a symmetric matrix in $\mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$) satisfies both of *Assumption 1* and *Assumption 3* since $h_{uv}(t) = \langle A(u+t(v-u))+b, v-u\rangle$ is linear and hence quasiconvex on $[0,1]$ for any $u,v \in \text{int}(\text{dom}(f)) = \mathbb{R}^n$.

From Examples 4.1 and 4.2, we see that the class of Problem (P) satisfying *Assumption 1* and *Assumption 3* is nonconvex in general. Subsequently, we propose an other version of Algorithm 3.1 that can be applied for such a kind of problems.

Algorithm 4.1 (NPG2)

Step 0 (Initialization). Select $t_0 > 0$, $0 < c_1 < c_0 < 1$, $x^0 \in \text{int}(\text{dom}(f))$, a tolerance $\varepsilon > 0$ and a positive real sequence $\{\gamma_k\}$ such that $\sum\limits_{k=0}^{+\infty} \gamma_k < \infty$. Taking $x^1 = \text{Prox}_{t_0 g}(x^0 - t_0\nabla f(x^0)), t_{-1} = t_0$ and $k = 1$.

Step 1.

$$\begin{aligned}
&\textbf{If} \quad \|\nabla f(x^k) - \nabla f(x^{k-1})\| > \frac{c_0}{t_{k-1}}\|x^k - x^{k-1}\| \\
&\textbf{then} \quad t_k = c_1 \frac{\|x^k - x^{k-1}\|}{\|\nabla f(x^k) - \nabla f(x^{k-1})\|} \qquad (4.1)\\
&\textbf{else} \quad \gamma'_{k-1} = \gamma_{k-1} \\
&\qquad \text{if } \frac{t_{k-1}}{t_{k-2}} < 1 \text{ then } \gamma'_{k-1} = \min\left\{\gamma_{k-1}, \sqrt{1+\frac{t_{k-1}}{t_{k-2}}} - 1\right\} \qquad (4.2)\\
&\qquad t_k = (1+\gamma'_{k-1})t_{k-1}.
\end{aligned}$$

Step 2. Compute $x^{k+1} = \text{Prox}_{t_k g}(x^k - t_k\nabla f(x^k))$.
Step 3. If $\|G_{1/t_k}(x^k)\| = \|x^k - x^{k+1}\|/t_k < \varepsilon$ **then** STOP **else** setting $k := k+1$ and return to **Step 1**.

Similar to the classical analysis under the nonconvex setting of f, we can show that for Algorithm 4.1, the norm of the gradient mapping converges to 0 and that all the limits points generated by the algorithm are stationary points of (P). We begin by presenting several preparatory lemmas.

Lemma 4.1 *The sequence $\{t_k\}$ in Algorithm 4.1 satisfies $\inf\limits_{k\geq 0} t_k > 0$ and has a positive limit, i.e., $\lim\limits_{k\to+\infty} t_k = t^* > 0$.*

Proof Similarly to Lemma 3.4 (i), it is clear that $t_k \geq \min\left\{t_0, \frac{c_1}{L_f}\right\} > 0$ for all $k \geq 0$. As a result, $\inf\limits_{k\geq 0} t_k > 0$. The remaining conclusion is shown as Lemma 3.4 (ii).

Lemma 4.2 *For Algorithm 4.1, there exists $\bar{k}$ such that*

$$\|\nabla f(x^k) - \nabla f(x^{k-1})\| \leq \frac{c_0}{t_{k-1}}\|x^k - x^{k-1}\| \ \ \forall k \geq \bar{k}.$$

Consequently, $0 < \inf\limits_{k\geq 0} t_k \leq t_{\bar{k}} \leq t_k \leq t_{k+1} \leq t^*$ for all $k \geq \bar{k}$.

Proof The proof is similar to that of Lemma 3.5.

The following lemma presents the sufficient decrease type inequality - a key step to obtain the convergence results of our algorithms.

Lemma 4.3 *Assuming that problem (P) satisfies Assumption 1 and Assumption 3 then the sequence $\{x^k\}$ generated by Algorithm 4.1 has the following property*

$$F(x^k) - F(x^{k+1}) \geq t_{\bar{k}}(1-c_0)\|G_{1/t^*}(x^k)\|^2, \ \forall k \geq \bar{k}.$$

Proof Invoking the Fundamental Theorem of Calculus, we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &= \int_0^1 \left\langle \nabla f(x^k + t(x^{k+1} - x^k)), x^{k+1} - x^k \right\rangle dt \\ &= \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \int_0^1 u_k(t)dt, \ \forall k \geq \bar{k}, \end{aligned} \tag{4.3}$$

where

$$\begin{aligned} u_k(t) &= \langle \nabla f(x^k + t(x^{k+1} - x^k)) - \nabla f(x^k), x^{k+1} - x^k \rangle \\ &= h_{x^k x^{k+1}}(t) - \langle \nabla f(x^k), x^{k+1} - x^k \rangle. \end{aligned}$$

According to *Assumption 3*, the quasiconvexity of $u_k(t)$ in $[0,1]$ follows that

$$\begin{aligned} u_k(t) &\leq \max\{u_k(0), u_k(1)\} = \max\{0, u_k(1)\} \leq |u_k(1)| \\ &= |\langle \nabla f(x^{k+1}) - \nabla f(x^k), x^{k+1} - x^k \rangle|, \ \forall t \in [0,1]. \end{aligned}$$

Thereafter, using Lemma 4.2, we derive that

$$\int_0^1 u_k(t)dt \leq \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2, \ \ \forall k \geq \bar{k}. \tag{4.4}$$

Now, combining (4.3), (4.4) and Lemma 2.5(ii) with $x = x^{k+1}$ we get that

$$\begin{aligned} F(x^k) - F(x^{k+1}) &= f(x^k) - f(x^{k+1}) + g(x^k) - g(x^{k+1}) \\ &\geq -\left\langle x^{k+1} - x^k, \nabla f(x^k) \right\rangle - \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2 + \left\langle x^{k+1} - x^k, \nabla f(x^k) + \frac{x^{k+1} - x^k}{t_k} \right\rangle \\ &= \frac{1-c_0}{t_k}\|x^{k+1} - x^k\|^2 \\ &\geq t_{\bar{k}}(1-c_0)\|G_{1/t_k}(x^k)\|^2 \qquad (4.5) \\ &\geq t_{\bar{k}}(1-c_0)\|G_{1/t^*}(x^k)\|^2, \quad \forall k \geq \bar{k}, \qquad (4.6) \end{aligned}$$

where the last inequality uses the monotonicity of the gradient mapping(Lemma 2.3) and the fact $t_{\bar{k}} \leq t_k \leq t^*$ from Lemma 4.2.

Theorem 4.1 *Under Assumptions 1 and 3, the following assertions hold for Algorithm 4.1:*

(i) The sequence $\{F(x^k)\}_{k \geq \bar{k}}$ is decreasing and for any $k \geq \bar{k}$, $F(x^{k+1}) < F(x^k)$ unless x^k is a stationary point of problem (P).

(ii) $\|G_{1/t^*}(x^k)\| \to 0$ as $k \to +\infty$.

(iii)

$$\min_{\bar{k} \leq k \leq K} \|G_{1/t^*}(x^k)\| \leq \min_{\bar{k} \leq k \leq K} \|G_{1/t_k}(x^k)\| \leq \sqrt{\frac{F(x^{\bar{k}}) - F_*}{t_{\bar{k}}(1-c_0)(K - \bar{k} + 1)}} = O\left(\frac{1}{\sqrt{K}}\right) \ \ \forall K \geq \bar{k}.$$

(iv) All limit points of the sequence $\{x^k\}_{k \geq \bar{k}}$ are stationary points of problem (P).

Proof (i) By (4.6) and $c_0 < 1$, it is clear to see that $F(x^k) \geq F(x^{k+1})$ for all $k \geq \bar{k}$. If $F(x^k) = F(x^{k+1})$ then $\|G_{1/t^*}(x^k)\| = 0$ meaning x^k is a stationary point of (P) (by Lemma 2.2).

(ii) Since problem (P) has a non-empty optimal solution set then the sequence $\{F(x^k)\}_{k\geq\bar{k}}$ is decreasing and lower bounded by F_*, moreover it is nonincreasing so it converges. Thus $F(x^k) - F(x^{k+1}) \to 0$. Combine with (4.6), we obtain $\|G_{1/t^*}(x^k)\| \to 0$ as $k \to +\infty$.

(iii) Summing (4.5) over $\bar{k}, ...K$ we get

$$F(x^{\bar{k}}) - F(x^{K+1}) \geq t_{\bar{k}}(1-c_0)\sum_{k=\bar{k}}^{K}\|G_{1/t^*}(x^k)\|^2 \geq t_{\bar{k}}(1-c_0)(K-\bar{k}+1)\min_{\bar{k}\leq k\leq K}\|G_{1/t_k}(x^k)\|^2$$
$$\geq t_{\bar{k}}(1-c_0)(K-\bar{k}+1)\min_{\bar{k}\leq k\leq K}\|G_{1/t^*}(x^k)\|^2.$$

The fact $F(x^{K+1}) \geq F_*$ completes the proof.

(iv) Let $\widehat{x}$ be a limit point of $\{x^k\}_{k\geq 0}$. Then there exists a subsequence $\{x^{k_i}\}$ which converges to $\widehat{x}$. From here for any $k_i > 0$,

$$\|G_{1/t^*}(\widehat{x})\| \leq \|G_{1/t^*}(\widehat{x}) - G_{1/t^*}(x^{k_i})\| + \|G_{1/t^*}(x^{k_i})\| \leq \left(\frac{2}{t^*} + L_f\right)\|\widehat{x} - x^{k_i}\| + \|G_{1/t^*}(x^{k_i})\|. \quad (4.7)$$

where the last inequality obtained from the Lipschitz continuity of the gradient mapping (Lemma 2.4). Since the right-hand side of (4.7) tends to 0 as $k_i \to +\infty$, we conclude that $G_{1/t^*}(\widehat{x}) = 0$, i.e., $\widehat{x}$ is a stationary point of Problem (P).

Remark 4.1 (i) Remember that $c_0, c_1 \in \left(0, \frac{1}{\sqrt{2}}\right)$ for Algorithm 3.1 (NPG1) but $c_0, c_1 \in (0,1)$ for Algorithm 4.1 (NPG2). This difference comes from the challenge of local Lipschitzness condition imposed on ∇f for NPG1. Intuitively, without the global Lipschitz condition of ∇f, the variation of ∇f with respect to x can be very large if we move a long step (which could be given by larger c_0, c_1) then the restriction of c_0, c_1 in $\left(0, \frac{1}{\sqrt{2}}\right)$ ensures the boundedness of the sequence $\{x^k\}$, thus avoiding the uncontrollable situation of the gradient. Theoretically, both NPG1 and NPG2 need to verify that the lower boundedness of t_k is a postive number; this can be derived from the global Lipschitzness of ∇f on $\overline{conv}\left(\{x^*, x^0, x^1 ...\}\right)$. Therefore, to use the locally Lipschitz gradient of f, NPG1 has to provide the compactness of $\overline{conv}\left(\{x^*, x^0, x^1, ...\}\right)$ that given by c_0, c_1 in $\left(0, \frac{1}{\sqrt{2}}\right)$. However, NPG2 works with the function f satisfying ∇f be globally Lipschitz then c_0, c_1 are just chosen to ensure the descent of the sequence of objective value $\{F(x^k)\}_{k\geq\bar{k}}$ as presented in (4.6).

(ii) Actually, the command (4.2) in Algorithm 4.1 is optional since we do not need it during the proof of the convergence of NPG2. However, through out the numerical experiments, we realize that this step improves the performance of the algorithm.

5 Problem (P) with quadratic function f

In this section, we propose an extension of NPG2 called *NPG-quad* solving Problem (P) with the quadratic function f, i.e., $f(x) = \frac{1}{2}x^T Ax + b^T x$ as described in Example 4.2. The changes compared with NPG2 are in the two points:

(i) Firstly,

$$t_k(\text{ of NPG-quad, in (5.2)}) = \frac{c_1\|x^k - x^{k-1}\|^2}{(x^k - x^{k-1})^T A(x^k - x^{k-1})} \geq \frac{c_1\|x^k - x^{k-1}\|}{\|Ax^k - Ax^{k-1}\|} = t_k(\text{ of NPG2, in (4.1)});$$

(ii) Secondly, c_0, c_1 in $(0,2)$ for NPG-quad while c_0, c_1 in $(0,1)$ for NPG2 . This extension stems from the new formula for t_k, as discussed above, and the special quadratic structure of f, which allows a better evaluation of $F(x^{k+1}) - F(x^k)$ as shown in (5.6) and (5.7).

These points probably make the stepsize of NPG-quad larger and therefore shorten the execution time compared to NPG1 and NPG2 .

Algorithm 5.1 (NPG-quad)

Step 0 (Initialization). Select $t_0 > 0$, $0 < c_1 < c_0 < 2$, $x^0 \in \text{dom}(g)$, a tolerance $\varepsilon > 0$ and a positive real sequence $\{\gamma_k\}$ such that $\sum_{k=0}^{+\infty} \gamma_k < +\infty$. Taking $x^1 = \text{Prox}_{t_0 g}(x^0 - t_0 \nabla f(x^0)), t_{-1} = t_0$, and $k = 1$.

Step 1.

$$\textbf{If} \quad (x^k - x^{k-1})^T A(x^k - x^{k-1}) > c_0 \frac{\|x^k - x^{k-1}\|^2}{t_{k-1}} \tag{5.1}$$

$$\textbf{then} \quad t_k = \frac{c_1 \|x^k - x^{k-1}\|^2}{(x^k - x^{k-1})^T A(x^k - x^{k-1})} \tag{5.2}$$

$$\textbf{else} \quad \gamma'_{k-1} = \gamma_{k-1}$$

$$\text{if } \frac{t_{k-1}}{t_{k-2}} < 1 \text{ then } \gamma'_{k-1} = \min\left\{\gamma_{k-1}, \sqrt{1 + \frac{t_{k-1}}{t_{k-2}}} - 1\right\} \tag{5.3}$$

$$t_k = (1 + \gamma'_{k-1}) t_{k-1}. \tag{5.4}$$

Step 2. Compute $x^{k+1} = \text{Prox}_{t_k g}(x^k - t_k \nabla f(x^k))$.
Step 3. If $\|G_{1/t_k}(x^k)\| = \|x^k - x^{k+1}\|/t_k < \varepsilon$ **then** STOP **else** setting $k := k+1$ and return to **Step 1**.

Lemma 5.1 *The sequence $\{t_k\}$ generated by Algorithm 5.1 has a positive limit, i.e., $\lim_{k \to +\infty} t_k = t^* > 0$..*

Proof Analogous to former sections, we are easy to have $t_k \geq \min\left\{t_0, \frac{c_1}{\|A\|}\right\} > 0$ for all $k \geq 0$. Therefore, $\inf_{k \geq 0} t_k > 0$. The computation of t_k by (5.2) or (5.4) provides $\ln\left(\frac{t_{k+1}}{t_k}\right) < \ln(1 + \gamma_k)$. The subsequent arguments are akin to the one of Lemma 3.4 (ii).

Lemma 5.2 *For Algorithm 5.1, there exists $\tilde{k}$ such that*

$$(x^k - x^{k-1})^T A(x^k - x^{k-1}) \leq c_0 \frac{\|x^k - x^{k-1}\|^2}{t_{k-1}}, \ \textit{for all } k \geq \tilde{k}. \tag{5.5}$$

Consequently, $0 < \inf_{k \geq 0} t_k \leq t_{\tilde{k}} \leq t_k \leq t_{k+1} \leq t^*$ for all $k \geq \tilde{k}$.

Proof Based on the properties of $\{t_k\}$ in Lemma 5.1 and arguing by contradiction as Lemma 3.5 we have the desired conclusion.

Theorem 5.1 *Suppose that Problem (P) satisfies Assumption 1 and f has a quadratic form as in Example 4.2. Let $\{x^k\}$ be the sequence generated by Algorithm 5.1. Then the sequence $\{F(x^k)\}_{k \geq \tilde{k}}$ is nonincreasing and $\|G_{1/t^*}(x^k)\| \xrightarrow{k \to +\infty} 0$. Additionally,*

$$\min_{\tilde{k} \leq k \leq K} \|G_{1/t^*}(x^k)\| \leq \min_{\tilde{k} \leq k \leq K} \|G_{1/t_k}(x^k)\| \leq \sqrt{\frac{F(x^{\tilde{k}}) - F_*}{t_{\tilde{k}}(1 - \frac{c_0}{2})(K - \tilde{k} + 1)}} = O\left(\frac{1}{\sqrt{K}}\right), \ \forall K \geq \tilde{k}$$

and any accumulation point of $\{x^k\}$ is a stationary point of (P).

Proof We have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &= \int_0^1 \left\langle \nabla f(x^k + t(x^{k+1} - x^k)), x^{k+1} - x^k \right\rangle dt \\ &= \int_0^1 \left\langle A(x^k + t(x^{k+1} - x^k)) + b, x^{k+1} - x^k \right\rangle dt \\ &= \left\langle A(x^{k+1} - x^k), x^{k+1} - x^k \right\rangle \int_0^1 t dt + \left\langle Ax^k + b, x^{k+1} - x^k \right\rangle \\ &= \frac{1}{2}(x^{k+1} - x^k)^T A(x^{k+1} - x^k) + \left\langle \nabla f(x^k), x^{k+1} - x^k \right\rangle. \end{aligned} \tag{5.6}$$

Now plugging (5.6) in $F(x^k) - F(x^{k+1})$ and using Lemma 2.5(ii) to obtain

$$\begin{aligned} F(x^k) - F(x^{k+1}) &= f(x^k) - f(x^{k+1}) + g(x^k) - g(x^{k+1}) \\ &\geq -\frac{1}{2}(x^{k+1} - x^k)^T A(x^{k+1} - x^k) - \langle \nabla f(x^k), x^{k+1} - x^k \rangle \\ &\quad + \left\langle x^{k+1} - x^k, \nabla f(x^k) + \frac{x^{k+1} - x^k}{t_k} \right\rangle \\ &= -\frac{1}{2}(x^{k+1} - x^k)^T A(x^{k+1} - x^k) + \frac{1}{t_k}\|x^{k+1} - x^k\|^2. \end{aligned} \tag{5.7}$$

Next, applying Lemma 5.2 for (5.7) we obtain for all $k \geq \tilde{k}$,

$$F(x^k) - F(x^{k+1}) \geq \left(1 - \frac{c_0}{2}\right) \frac{\|x^{k+1} - x^k\|^2}{t_k} = t_k \left(1 - \frac{c_0}{2}\right) \|G_{1/t_k}(x^k)\|^2 \geq t_{\tilde{k}} \left(1 - \frac{c_0}{2}\right) \|G_{1/t_k}(x^k)\|^2 \tag{5.8}$$

$$\geq t_{\tilde{k}} \left(1 - \frac{c_0}{2}\right) \|G_{1/t^*}(x^k)\|^2 \tag{5.9}$$

The remaining arguments are similar to those of Theorem 4.1.

Remark 5.1 If f is a concave quadratic function i.e., A is negative semi-definite then the condition (5.1) is false, hence

- $\tilde{k}$ in Lemma 5.2 should be zero;
- t_k is always defined by formula (5.4) and $\{t_k\}_{k\geq 0}$ is increasing to a finite limit;
- the evaluation (5.8) should be

$$F(x^k) - F(x^{k+1}) \geq \frac{\|x^{k+1} - x^k\|^2}{t_k}, \ \forall k \geq 0. \tag{5.10}$$

6 Numerical experiments

In this section, we investigate the performance of our new stepsize for the proximal gradient scheme by comparing NPG1 (Algorithm 3.1), NPG2 (Algorithm 4.1) and NPG-quad (Algorithm 5.1) with the recent algorithms including:

- the AdPG proposed by Malitksy and Mishchenko [24] (Algorithm 3 in [24]);
- the AdaPG^{q,r} from Latafat et al. in [19] using $(q,r) = \left(\frac{3}{2}, \frac{3}{4}\right)$;
- the proximal gradient algorithms with stepsize selection based on an improved version of Armijo's backtracking procedure[1], denoted by PG-LS(s,r) where (s,r) equals $(1.1, 0.5)$ or $(1.2, 0.5)$.

[1] For $s > 1$, $r < 1$, Armijo's line search in finds the largest $t_k = sr^i t_{k-1}$ for $i = 0, 1, \ldots$ such that $f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{1}{2t_k}\|x^{k+1} - x^k\|^2$.

To ensure fairness, all of the parameters chosen for PG-LS and AdaPG^{q,r} are the ones with the most stable and efficient empirical performance reported in [24], [19].

For our algorithms, we use the convergent series $\sum_{k=0}^{+\infty} \gamma_k$ defined by

$$\gamma_{k-1} = \frac{0.1(\ln k)^{5.7}}{k^{1.1}}, \quad \forall k \geq 1, \tag{6.1}$$

and $(c_0, c_1) = (0.7, 0.69)$ for NPG1 , $(c_0, c_1) = (0.99, 0.98)$ for NPG2 and NPG-quad .

Some discussions on parameters setting.

The parameter choices for the NPG algorithms are guided by a simple yet consistent intuition: choose values that allow the stepsize to be as large as possible while maintaining algorithmic stability. Specifically:

- The first factor that affects the magnitude of our stepsize is (c_0, c_1). For NPG1 and NPG2, we select (c_0, c_1) to be as close as possible to their theoretical limits—$(0.7, 0.69)$ and $(0.99, 0.98)$, respectively. This maximizes the stepsize without violating convergence guarantees.
- In the case of NPG-quad, despite having a much larger allowable range for (c_0, c_1), we observed that choosing values that are too large can lead to unstable behavior (sometimes resulting in surprisingly fast convergence for problems requiring tens of thousands of iterations, but often leading to poor performance on easier problems that need only a few hundred iterations). To ensure consistent performance across problem scales, we again choose $(0.99, 0.98)$ as a balanced and robust setting.
- The second factor that helps increase the stepsize is the sequence γ_k, a natural choice to satisfy our assumption is $\frac{1}{k^p}$ with p close to 1. We could further enlarge this sequence by changing the numerator adaptively, for example [22] proposed $\gamma_k = \frac{w_k}{k^p}$ with w_k chosen adaptively. We instead opt for the explicit form $\gamma_k = \frac{a(\ln k)^b}{k^p}$ similar to [16]. Stick to our guiding intuition, we choose p close to 1, b large to generate gradually large γ_k and a small to ensure stability. This choice also eliminates the need for computing w_k while increasing the stepsize effectively and leads to empirical improvements.

Test problems

We conduct experiments on five typical composite type optimization problems with various sizes for each one. The average results on 10 randomly generated data for each size of considered problems are reported with respect to

1. the number of iterations (*Iter.*);
2. $\|G_{1/t_k}(x^k)\| = \|x^{k+1} - x^k\|/t_k$ (*Res.*);
3. $F(x^k) - F_*$ (*Obj.*), where F_* is computed as the minimum of $F(x^k)$ over all iterations and all tested algorithms;
4. the running time in seconds (*Time(s)*).

For all implemented algorithms, the stopping criterion is either the residual $\|G_{1/t_k}(x^k)\| \leq 1e-06$ or if the maximum of N_{max} iterations is reached. The detailed information is on Tables 1, 2, 3, 4, 5. We emphasize the best results among all by bold characters and the worst results by italic type. We also choose one arbitrary data for each kind of problems to illustrate the performance by Figures 1, 2, 3, 4, 5.

All experiments[2] were implemented in Python and executed on a computer equipped with a 12th Gen Intel(R) Core(TM) i7-1260P 2.10 GHz processor.

[2] All codes are available at our repository https://github.com/hoaiphamthi/NPG-for-composite-models.

Summary on the experimental results

In all problems considered, NPG1 and NPG2 notably outperform other algorithms in every criterion, except for the experiments on the Lasso problem where NPG-quad yields superior performance. This advantage stems from the ability of these algorithms to produce suitably large stepsizes . The detailed progression and overall distribution (after removing outliers) of these stepsizes are shown in Figure 1.

6.1 Lasso problems

The formulation of Lasso problem is formulated as the ℓ_1 regularized least squares

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1, \tag{Lasso}$$

where $A\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^m$. The applications of Lasso can be found in statistic, machine learning, signal processing, see e.g., [3,14]. By using the similar rules in [14], we randomly generate $A\in\mathbb{R}^{m\times n}$ with entries drawn from the normal distribution $\mathcal{N}(0,1)$. We then construct a sparse solution x^* with 5% approximately non-zero entries, drawn from a mixture distribution $\mathcal{N}(0,1)\times B(1,0.05)$ then setting $b=Ax^*+\delta$, where δ is white Gaussian noise with variance 0.01. The regularization term $\lambda=0.01\|A^Tb\|_\infty$. Obviously, Lasso satisfies *Assumptions 1, 2, 3* then both of NPG1 and NPG2 are available for it. Moreover, f is quadratic hence NPG-quad can be applied for solving this problem formally. Figure 1 illustrates the performance of mentioned algorithms for one of randomly generated data with $m=2048, n=8192$. The obtained average results in Table 1 show the best performance of NPG-quad for almost dimensions of Lasso.

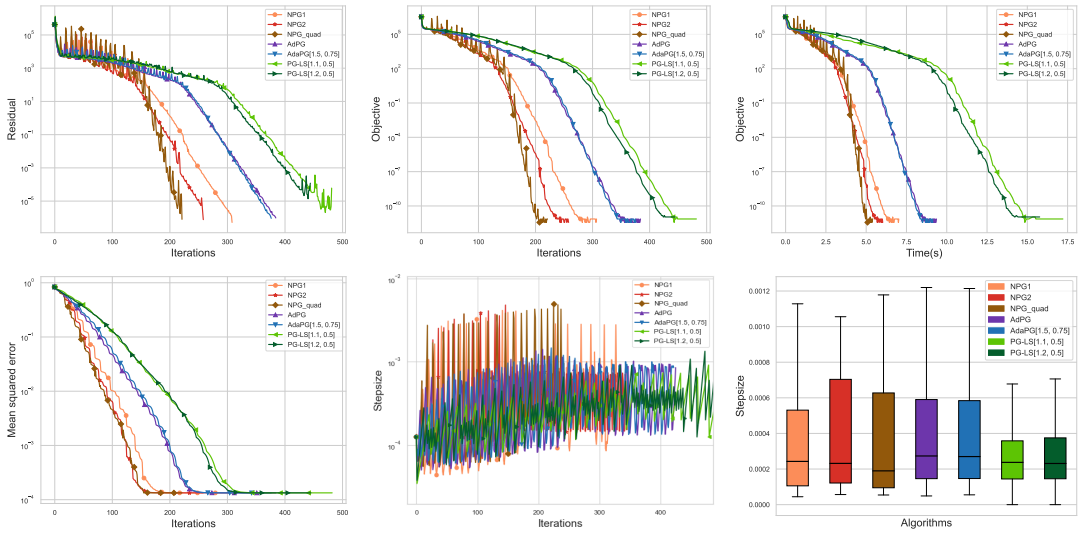

Fig. 1: Lasso problem with $m=2048, n=8192$.

6.2 Minimum length piecewise-linear curve subject to equality constraints

We consider an other optimization problem from [10, Example 10.4], where the objective is minimizing the length of the piecewise-linear curve connecting the points $(0,0),(1,x_1),\ldots,(n,x_n)$ such that $Ax=b$, where $A\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^m$. The problem therefore can be formed as

$$\min \sqrt{1+x_1^2}+\sum_{i=1}^{n-1}\sqrt{1+(x_{i+1}-x_i)^2} \quad \text{s.t.} \quad Ax=b. \tag{Min-length}$$

Size		Metrics	Average of all datasets						
m	n		NPG1	NPG2	NPG-quad	AdPG	AdaPG$^{(\frac{3}{2},\frac{3}{4})}$	PG-LS (1.1, 0.5)	PG-LS (1.2, 0.5)
512	1024	Iter.	112,7	104	**87,3**	125,5	126,4	*162*	150,9
		Res.	7,55E-07	8,16E-07	5,72E-07	*8,36E-07*	7,78E-07	5,73E-07	**4,09E-07**
		Obj.	**9,09E-14**	1,08E-13	1,02E-13	1,14E-13	*1,25E-13*	9,66E-14	1,14E-13
		Time(s)	0,047803	0,039699	**0,036495**	0,051732	0,056574	*0,088408*	0,087751
512	2048	Iter.	285,6	238,2	**228,9**	365,1	358,3	*467*	459,5
		Res.	8,41E-07	7,53E-07	7,48E-07	*9,12E-07*	9E-07	**2,33E-07**	3,69E-07
		Obj.	**1,93E-13**	**1,93E-13**	2,5E-13	**1,93E-13**	2,73E-13	3,52E-13	*4,55E-13*
		Time(s)	0,13425	**0,120803**	0,135755	0,244074	0,272553	0,423316	*0,487806*
512	4096	Iter.	*14194*	12998,6	**8687,2**	13984,2	13523,8	13872,8	13775,7
		Res.	6,37E-05	1,51E-05	**9,9E-07**	3,05E-05	1,56E-05	*0,000149*	0,000135
		Obj.	4,05E-09	1,96E-10	**4,32E-13**	2,98E-09	6,96E-10	*6,73E-08*	5,97E-08
		Time(s)	20,86881	18,70222	**11,56859**	18,75128	19,17551	20,96922	*23,0043*
1024	2048	Iter.	118,3	109,3	**97,5**	132	135,8	*184,4*	161,9
		Res.	*8,43E-07*	7,57E-07	**3,92E-07**	7,61E-07	7,42E-07	4,55E-07	5,24E-07
		Obj.	4,55E-13	4,09E-13	**3,18E-13**	*4,77E-13*	4,32E-13	**3,18E-13**	4,32E-13
		Time(s)	0,102231	**0,098784**	0,103587	0,143296	0,164256	0,310328	*0,31076*
1024	4096	Iter.	271,9	226,4	**219**	338,1	334,8	*442*	416,1
		Res.	9E-07	9,1E-07	5,74E-07	9,08E-07	*9,27E-07*	2,78E-07	**1,93E-07**
		Obj.	**1E-12**	1,27E-12	1,23E-12	1,27E-12	1,05E-12	*2,05E-12*	1,77E-12
		Time(s)	1,007017	**0,880541**	0,883184	1,299208	1,249578	1,950749	*2,153433*
1024	8192	Iter.	*14642,7*	13684,4	**9785,1**	14341,5	14121,1	13911,2	13815,6
		Res.	9,16E-05	0,00011	**1,17E-06**	7,14E-05	7,01E-05	*0,000364*	0,000267
		Obj.	2,88E-08	3,93E-09	**2,18E-12**	1,63E-08	6,83E-09	*2,25E-07*	1,9E-07
		Time(s)	121,7695	113,761	**82,05926**	119,6346	120,2897	139,2143	*151,4469*
2048	4096	Iter.	115,5	106,3	**88,5**	120,4	124,9	*164*	155,5
		Res.	7,61E-07	*7,88E-07*	5,03E-07	7,28E-07	7,84E-07	4,98E-07	**4,61E-07**
		Obj.	**1,36E-12**	*2,18E-12*	1,82E-12	2E-12	1,55E-12	1,82E-12	1,55E-12
		Time(s)	1,03452	1,022059	**0,827298**	1,104417	1,152975	1,96307	*2,050739*
2048	8192	Iter.	291,1	260,3	**216,5**	361,4	356,7	*472*	444,5
		Res.	8,12E-07	9,06E-07	5,98E-07	*9,38E-07*	9,13E-07	**0**	**0**
		Obj.	4,37E-12	4,73E-12	5,09E-12	4,37E-12	**3,27E-12**	6,18E-12	*8,73E-12*
		Time(s)	5,30504	4,788649	**3,924855**	6,439194	6,361541	10,36916	*10,58561*

Table 1: Average results for Lasso problem

It is seen that Min-length[3] satisfies *Assumptions 1,2,3* and we can use NPG1 and NPG2 to solve it exactly. In the implementation, all members of A are randomly generated by normal distribution $\mathcal{N}(0,1)$. Taking $b = Ax^*$, where $x^* \sim \mathcal{N}(0,1)$. Figure 2 provides the line graphs of one randomly generated data with $m = 100, n = 10000$. Table 2 includes the average computation results for various sizes of Min-length problem. Notably, both NPG1 and NPG2 outperform the remaining ones with the big deviation in term of computational time, residual, objective value and the number of iterations. The speed of NPG1 can be seen as the best among all for Min-length.

[3] Min-length is a case of problem (P) with $f(x) = \sqrt{1+x_1^2} + \sum_{i=1}^{n-1} \sqrt{1+(x_{i+1}-x_i)^2}$ and $g(x) = \imath_C$ (the indicator function of C) with $C = \{x \in \mathbb{R}^n \mid Ax = b\}$.

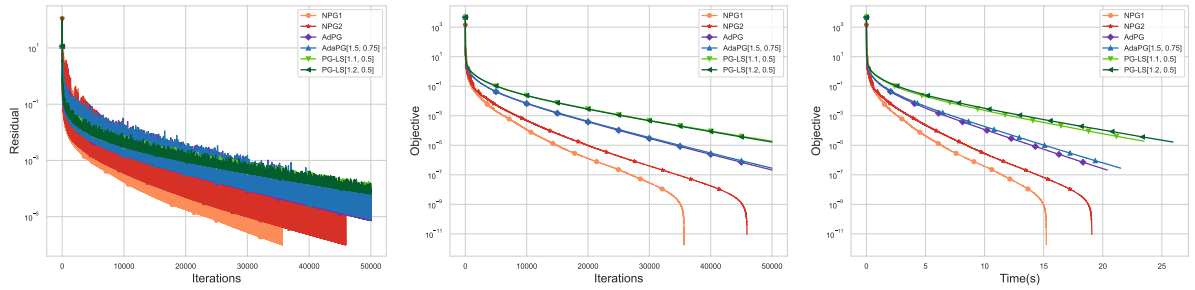

Fig. 2: Min-length problem with $m = 100, n = 10000$.

Size		Metrics	Average of all datasets					
m	n		NPG1	NPG2	AdPG	AdaPG$^{(\frac{3}{2},\frac{3}{4})}$	PG-LS (1.1, 0.5)	PG-LS (1.2, 0.5)
50	5000	Iter.	**30356,2**	38975,3	49876,5	49992	*50000*	*50000*
		Res.	**9,99E-07**	1E-06	5,18E-06	5,34E-06	3,65E-05	*3,9E-05*
		Obj.	**1,46E-12**	3,45E-11	1,07E-07	1,37E-07	*7,22E-06*	6,76E-06
		Time(s)	**7,505386**	9,706766	14,3997	16,53744	17,63717	*19,29526*
500	5000	Iter.	490,7	**439,8**	1162,3	1186,8	*1500*	*1500*
		Res.	**9,4E-07**	9,46E-07	9,84E-07	9,81E-07	8,52E-06	*8,84E-06*
		Obj.	**4,55E-13**	7,73E-12	1,18E-11	1,08E-11	3,76E-09	*4,25E-09*
		Time(s)	0,665004	**0,588654**	1,430408	1,466578	2,037316	*2,274157*
2000	5000	Iter.	**71,7**	89,2	127,9	127,2	*188,6*	184,7
		Res.	7,31E-07	**6,11E-07**	9,33E-07	9,59E-07	*27,83445*	4,102704
		Obj.	**7,28E-13**	1,36E-12	2,36E-12	2,82E-12	*5E-12*	4,46E-12
		Time(s)	**0,729847**	0,792776	0,987844	0,962354	1,768878	*1,802494*
100	10000	Iter.	**36416,5**	46205,7	*50000*	*50000*	*50000*	*50000*
		Res.	**9,99E-07**	1,74E-06	9,63E-06	1,87E-05	7,63E-05	*8,1E-05*
		Obj.	**1,27E-12**	1,35E-09	4,18E-07	5,38E-07	*2,68E-05*	2,49E-05
		Time(s)	**17,90613**	22,93893	25,18967	27,82387	29,32302	*33,04613*
1000	10000	Iter.	496,1	**447,4**	1184,1	1215,6	*1500*	*1500*
		Res.	9,57E-07	**9,34E-07**	9,79E-07	9,93E-07	*9,32E-06*	7,88E-06
		Obj.	**2,18E-12**	6,37E-12	1,13E-11	1,26E-11	*3,83E-09*	3,08E-09
		Time(s)	3,567836	**2,937245**	7,600918	7,894068	11,02318	*12,24045*
2000	10000	Iter.	161,8	**154,6**	364,3	375,4	*500*	*500*
		Res.	**8,19E-07**	8,78E-07	9,42E-07	9,73E-07	*8,3E-06*	3,5E-06
		Obj.	4,91E-12	**4,37E-12**	8,37E-12	9,46E-12	*1,4E-09*	1,64E-10
		Time(s)	2,512865	**2,218866**	5,000347	5,190868	7,848265	*8,723222*

Table 2: Average results for Min-length problem ($N_{max} = 50000$).

6.3 Dual of the entropy maximization problems

We consider the entropy maximization problem subject to linear constraints [10, Section 5.1.6] which is

$$\min \sum_{i=1}^{n} x_i \log x_i \quad \text{s.t.} \quad Ax \leq b, \ \sum_{i=1}^{n} x_i = 1, \ \text{and} \ x_i > 0, i = 1, ..., n, \tag{6.2}$$

where $A = [a^1, a^2, ..., a^n] \in \mathbb{R}^{m\times n}$, with $a^i \in \mathbb{R}^m$ is the $i-$th column of A and $b \in \mathbb{R}^m$. Its dual problem is

$$\min e^{-\mu-1} \sum_{i=1}^{n} e^{-(a^i)^T\lambda} + b^T\lambda + \mu, \ \text{s.t.} \ \lambda \in \mathbb{R}^m_+, \mu \in \mathbb{R}. \tag{Dual-max-entropy}$$

It is observed that Problem Dual-max-entropy[4] matches *Assumptions 1, 2* but *Assumption 3*. Therefore the use of NPG1 is straightforward for it. We still run NPG2 for Dual-max-entropy as a heuristic approach. We use the similar rule of generating data as [24]. Specifically, a $m \times n$ matrix A with entries are generated from $\mathcal{N}(0,1)$, $b = Ax^*$ with a ℓ_1-normalized x^* sampled from the uniform distribution $\mathcal{U}[0.1,1)$. Results are depicted in Table 3 and Figure 3. It is shown that the performance of NPG2 is more significantly efficient than the remaining ones.

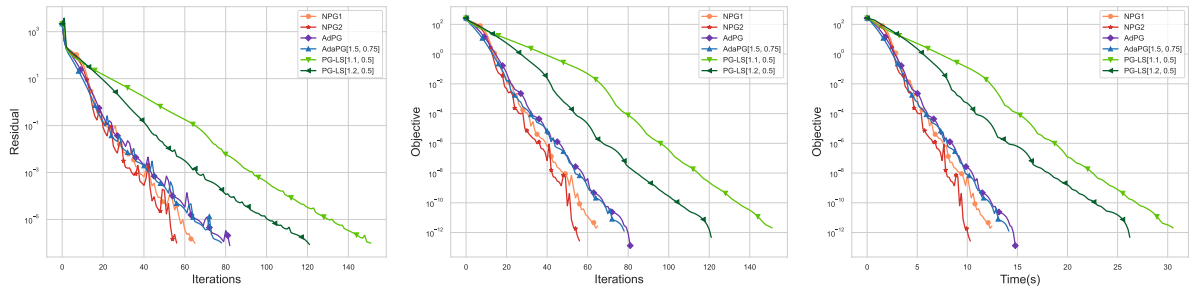

Fig. 3: Dual-max-entropy problem with $m = 4000, n = 5000$.

Size		Metrics	Average of all datasets					
m	n		NPG1	NPG2	AdPG	AdaPG$^{(\frac{3}{2},\frac{3}{4})}$	PG-LS (1.1, 0.5)	PG-LS (1.2, 0.5)
100	500	Iter.	31,1	**28,9**	33	31,9	*79,9*	50,9
		Res.	5,94E-07	5,26E-07	**4,62E-07**	6,18E-07	*6,91E-07*	6,48E-07
		Obj.	7,35E-14	5,79E-14	**2,76E-14**	*9,33E-14*	8,31E-14	7,66E-14
		Time(s)	0,019625	**0,019314**	0,020991	0,020471	*0,047429*	0,02965
500	2000	Iter.	34,2	**31,8**	35,4	33,3	*84,5*	54,6
		Res.	7,25E-07	**6,13E-07**	6,69E-07	7,03E-07	*8,74E-07*	7,15E-07
		Obj.	1,62E-13	**5,92E-14**	1,29E-13	1,18E-13	*4,04E-13*	1,78E-13
		Time(s)	0,379643	**0,348487**	0,395345	0,357533	*0,9504*	0,657475
2000	4000	Iter.	49,4	**45,9**	49,1	49	*102,3*	70,5
		Res.	**7,67E-07**	7,87E-07	8,24E-07	7,92E-07	*8,65E-07*	8,6E-07
		Obj.	**2,99E-13**	*2,92E-12*	4,8E-13	4,23E-13	5,86E-13	4,68E-13
		Time(s)	4,154221	3,604547	3,701799	**3,592433**	*7,761822*	5,752356
4000	5000	Iter.	76,4	**61**	82,7	79,2	*154,5*	119,6
		Res.	7,88E-07	**7,2E-07**	8,18E-07	9,07E-07	*9,25E-07*	8,96E-07
		Obj.	5,94E-13	**4,14E-13**	1,44E-12	1,7E-12	*2,05E-12*	1,77E-12
		Time(s)	14,33235	**10,97741**	15,10879	14,14594	*31,60767*	26,71427

Table 3: Average results for Dual-max-entropy problem.

6.4 Maximum likelihood estimate of the information matrix

This problem (see [10, Eq. (7.5)]) aims to estimate the inverse of a covariance matrix Y of a multivariate random variable subject to the eigenvalue bounds given some samples of the random variable. The problem can be formulated as

$$\min \quad f(X) = -\log\det(X) + \operatorname{tr}(XY) \quad \text{s.t.} \quad X \in \mathbb{S}_n \ \text{ and } \ lI \preceq X \preceq uI. \qquad \text{(Max-likelyhood)}$$

Here $\mathbb{S}_n$ denotes the space of real symmetric matrices of dimension $n \times n$, and $A \preceq B$ indicates that $B - A$ is positive semi-definite. Observably, Max-likelyhood[5] satisfies *Assumptions 1,2,3* then

[4] Dual-max-entropy is a case of problem (P) with $f(\lambda,\mu) = e^{-\mu-1}\sum_{i=1}^{n} e^{-(a^i)^T\lambda} + b^T\lambda + \mu$ and $g(\lambda,\mu) = \iota_C$ (the indicator function of C) with $C = \mathbb{R}^m_+ \times \mathbb{R}$ and ∇f is not globally Lipschitz on C.

[5] Max-likelyhood is a case of problem (P) with $f(X) = -\log\det(X) + \operatorname{tr}(XY)$ and $g(X) = \iota_C$ (the indicator function of C) with $C = \{X \in \mathbb{S}_n \mid lI \preceq X \preceq uI\}$.

NPG1 and NPG2 are exact methods to solve Max-likelyhood. The dataset for the implementation is generated analogously to [24] as follows. We initially generate a random vector $y \in \mathbb{R}^n$ with entries from $\mathcal{N}(0,10)$ and $\delta_i \in \mathbb{R}^n$ with entries from $\mathcal{N}(0,1)$, and then set $y_i = y + \delta_i$, $i = 1,\ldots,M$. The covariance matrix of the samples $y_1,\ldots,y_M$ is $Y = \frac{1}{M}\sum_{i=1}^{M} y_i y_i^T$. The obtained results are shown in Table 4 and Figure 4. It can be seen that for the Max-likelyhood problem, both of NPG1 and NPG2 provide better results compared to the others with the big deviation. And most of cases NPG2 performs best.

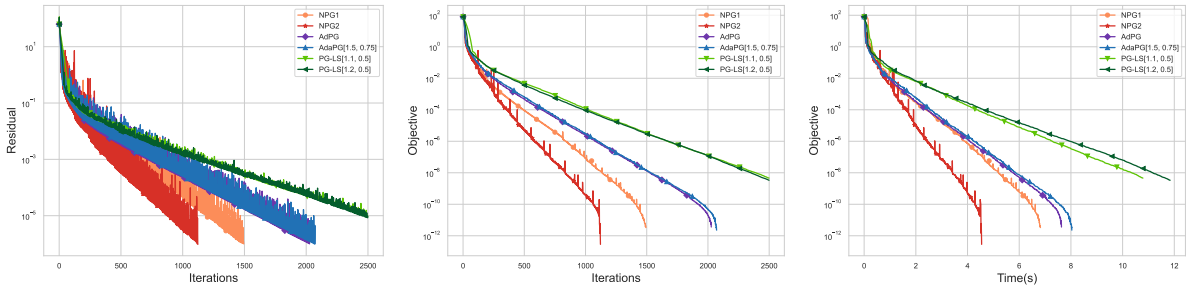

Fig. 4: Max-likelyhood problem with $n = 50, l = 0.1, u = 1000, M = 100$.

Size	Metrics	Average of all datasets					
n, l, u, M		NPG1	NPG2	AdPG	AdaPG$^{(\frac{3}{2},\frac{3}{4})}$	PG-LS (1.1, 0.5)	PG-LS (1.1, 0.5)
100, 0.1, 10, 50	Iter.	1822,4	**1556,6**	2339,4	2398,9	*3000*	*3000*
	Res.	**9,83E-07**	9,86E-07	9,84E-07	9,9E-07	*21,57575*	11,36214
	Obj.	6,98E-12	**6,23E-12**	6,88E-12	7,8E-12	*4,85E-09*	3,89E-09
	Time(s)	36,29822	**32,67365**	49,44176	51,19477	71,15808	*78,05057*
100, 0.1, 10, 500	Iter.	111,5	**97,7**	146,8	148,3	*200*	*200*
	Res.	8,59E-07	**7,64E-07**	8,75E-07	8,97E-07	1,33E-05	*1,832449*
	Obj.	1,86E-12	**1,36E-12**	2,05E-12	2,59E-12	*5,34E-10*	5,73E-11
	Time(s)	2,487496	**2,018361**	2,991299	2,986011	4,602473	*5,2777*
100, 0.1, 10, 1000	Iter.	60,2	**50,9**	59,4	57,9	*100*	95,6
	Res.	7,28E-07	**6,49E-07**	8,98E-07	6,82E-07	3,75E-06	*1,11606*
	Obj.	**1,14E-12**	1,22E-12	2,02E-12	3,92E-12	*3,15E-11*	2,41E-12
	Time(s)	1,316086	**1,071492**	1,172037	1,126762	2,087067	*2,727355*
30, 0.1, 1000, 50	Iter.	3887,1	**3126,7**	4222,1	4258	*5000*	*5000*
	Res.	0,000121	**3,72E-05**	0,000173	0,000188	*9,037132*	6,382411
	Obj.	4,15E-05	**8,47E-13**	0,000104	0,000116	0,000564	*0,000586*
	Time(s)	5,137378	**4,249085**	6,01792	6,145084	7,533717	*8,229293*
50, 0.1, 1000, 100	Iter.	1498,7	**1147,9**	1959,1	1981,8	*2500*	*2500*
	Res.	9,64E-07	**9,64E-07**	2,66E-06	1,93E-06	*5,883939*	5,688969
	Obj.	**1,78E-12**	4,21E-12	8,99E-11	1,26E-10	*7,83E-08*	6,93E-08
	Time(s)	5,882136	**4,330707**	7,377842	7,466817	10,49909	*11,67578*

Table 4: Average results for Max-likelyhood problem.

6.5 Nonnegative matrix factorization

One of efficient approaches to solve recommendation system problems [27] is based on nonnegative matrix factorization[6]

$$\min f(U,V) = \frac{1}{2}\|UV^T - A\|_F^2, \quad \text{s.t. } U \in {\mathbb{R}_+}^{m\times r}, V \in {\mathbb{R}_+}^{n\times r}, \tag{NMF}$$

[6] NMF is a case of problem (P) with $f(U,V) = \frac{1}{2}\|UV^T - A\|_F^2$ and $g(U,V) = \imath_C$ (the indicator function of C) with $C = {\mathbb{R}_+}^{m\times r} \times {\mathbb{R}_+}^{n\times r}$.

where $A \in \mathbb{R}^{m\times n}$ is a low-rank matrix, $\|\cdot\|_F$ stands for Frobenius norm. This problem does not satisfy *Assumption 2* and *Assumption 3*. Therefore our algorithms can be seen as heuristic methods for it. Akin to [24], we create A by multiplying matrices B and $C^\top$, where $B \in \mathbb{R}_+^{m\times r}$ and $C \in \mathbb{R}_+^{n\times r}$ have entries drawn from a normal distribution $\mathcal{N}(0,1)$. All negative entries of B and C are replaced with zero. The numerical results are reported in Table 5 and illustrated by Figure 5. For this problem, NPG1 and NPG2 alternately are proved to be the most effective methods compared to the others.

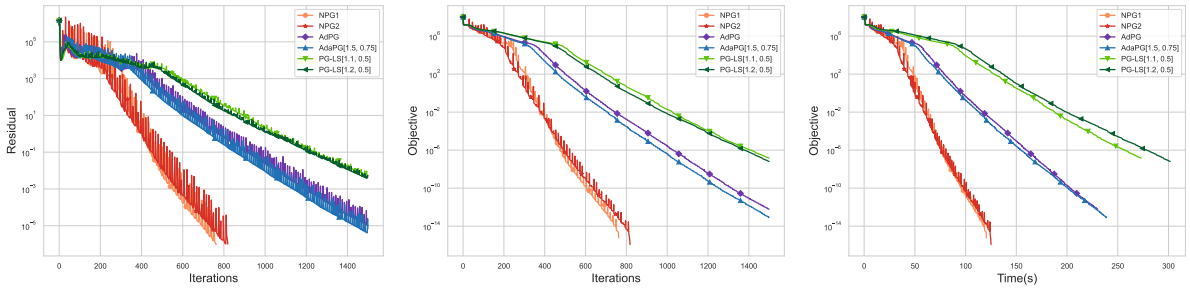

Fig. 5: NMF problem with $m = 3000, r = 30, n = 3000$.

7 Conclusions

In this paper, we propose an efficient explicit stepsize applied for the proximal gradient (PG) scheme. In particular, Algorithm 3.1 (NPG1) solves the convex situation of the problem (P) under the locally Lipschitz gradient condition imposed on f. The iterates is proved to converge to an optimal solution of (P) with the computational complexity $O\left(\frac{1}{k}\right)$ of $F(x^k) - F_*$ and the Q-linear rate if f has locally strong convexity property. These convergence results are based on the descent of our proposed method. Moreover, our stepsize is also investigated with a class of nonconvex f satisfying global Lipschitz gradient condition with Algorithm 4.1 (NPG2), where the size of step length can be bigger. In quadratic case of f, Algorithm 5.1 (NPG-quad) is improved significantly in length of stepsizes for solving (P). Basically, our stepsize selection is computed quickly by a closed formulas without line search computation or estimating some constant (like Lipschitz constant of gradient) to ensure the convergence of the PG algorithms. Moreover, the increasing of the sequence of our stepsizes from some fixed iteration opens the ability to speed up the corresponding PG algorithms. The deep experiments on a variety of test instances with various sizes show the crucial efficiency of the proposed method compared to the recent ones. Future research includes deploying our adaptive stepsize for the composite models in the absence of both convexity and global Lipschitz gradient assumptions on f.

References

1. M. Ahookhosh, A. Themelis, and P. Patrinos. A bregman forward-backward linesearch algorithm for nonconvex composite optimization: superlinear convergence to nonisolated local minima. *SIAM J. Optim.*, 31(1):653–685, 2021.
2. H.H. Bauschke, J. Bolte, and M. Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.
3. A. Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. Society for Industrial and Applied Mathematics, USA, 2014.
4. A. Beck. *First Order Methods in Optimization*. Society for Industrial and Applied Mathematics, USA, 2017.
5. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2:183–202, 2009.

Size			Metrics	Average of all datasets					
m	r	n		NPG1	NPG2	AdPG	AdaPG$^{(\frac{3}{2},\frac{3}{4})}$	PG-LS (1.1, 0.5)	PG-LS (1.1, 0.5)
500	20	1000	Iter.	456,3	**453,5**	860,5	823,9	*1000*	996,8
			Res.	**9,15E-07**	9,41E-07	9,7E-07	9,71E-07	*9,25E-05*	3,77E-05
			Obj.	3,47E-15	**2,75E-15**	6,29E-15	7,24E-15	*2,62E-10*	3,6E-11
			Time(s)	**4,156899**	4,471811	7,8527	8,130274	9,781315	*11,24854*
1000	20	500	Iter.	**455,4**	468,9	864,8	836,2	*1000*	*1000*
			Res.	9,28E-07	**9,14E-07**	9,82E-07	9,52E-07	*7,32E-05*	4,35E-05
			Obj.	1,86E-15	**8,76E-16**	7,23E-15	6,22E-15	*2,17E-10*	5,39E-11
			Time(s)	**3,302716**	3,437761	6,337801	6,548401	8,721143	*9,45892*
2000	20	3000	Iter.	**465,8**	475,9	834,4	810,2	*1000*	*1000*
			Res.	9,33E-07	**9,1E-07**	9,73E-07	9,59E-07	*9,69E-05*	3,27E-05
			Obj.	7,76E-16	**1,94E-16**	2,35E-15	2,27E-15	*2,83E-10*	1,41E-11
			Time(s)	**32,64289**	33,53955	57,29892	58,51258	84,21695	*91,36899*
3000	20	2000	Iter.	**456,8**	473,8	835,7	794,9	*1000*	995,8
			Res.	9,74E-07	**9,04E-07**	9,69E-07	9,77E-07	*6,96E-05*	1,6E-05
			Obj.	9,5E-16	**1,94E-16**	1,9E-15	2,08E-15	*4,31E-11*	3,53E-12
			Time(s)	**36,32393**	37,87897	66,04371	64,52914	95,5099	*106,5964*
3000	20	3000	Iter.	**423,2**	426,2	801,5	762	*1000*	998,6
			Res.	**9,05E-07**	9,1E-07	9,79E-07	9,38E-07	*1,27E-05*	6,18E-06
			Obj.	5,93E-16	**1,12E-16**	1,47E-15	1,22E-15	*1,11E-12*	1,98E-13
			Time(s)	**48,40132**	48,81449	90,80086	88,1426	129,2444	*145,173*
500	30	1000	Iter.	1064,6	**994,7**	*1500*	1496,3	*1500*	*1500*
			Res.	9,8E-07	**9,56E-07**	5,88E-05	1,46E-05	*0,005371*	0,002949
			Obj.	2,06E-15	**1,42E-15**	3,16E-11	1,29E-11	*9,98E-07*	3,31E-07
			Time(s)	9,95874	**8,66614**	13,01264	14,02261	16,1783	*18,08454*
1000	30	500	Iter.	988,2	**974,1**	*1500*	1498,6	*1500*	*1500*
			Res.	9,66E-07	**9,33E-07**	4,08E-05	4,33E-05	*0,008548*	0,003825
			Obj.	3,47E-15	**8,97E-16**	2,73E-11	4,24E-11	*4,35E-06*	6,99E-07
			Time(s)	8,499324	**8,324084**	13,13515	13,40158	13,96775	*15,15622*
2000	30	3000	Iter.	**712,6**	749,8	1464,7	1457,5	*1500*	*1500*
			Res.	**9,6E-07**	9,77E-07	4,57E-06	2,25E-06	*0,001066*	0,000886
			Obj.	**3,49E-16**	6,77E-16	8,72E-14	3,06E-14	*1,39E-08*	9,9E-09
			Time(s)	**55,57149**	56,35817	111,9849	113,7706	129,9203	*143,5925*
3000	30	2000	Iter.	**710,6**	748,6	1484,3	1447,1	*1500*	*1500*
			Res.	**9,55E-07**	9,59E-07	3,97E-06	1,71E-06	0,001185	*0,001274*
			Obj.	1,28E-15	**4,85E-16**	2,79E-13	2,47E-14	1,22E-08	*1,92E-08*
			Time(s)	**57,19699**	63,18712	122,0072	124,4837	143,6102	*158,8972*
3000	30	3000	Iter.	**714,8**	794,3	1465,6	1462,3	*1500*	*1500*
			Res.	**9,59E-07**	9,72E-07	1,24E-05	3,7E-06	*0,002207*	0,001807
			Obj.	7,04E-16	**4,25E-16**	6,68E-13	2,79E-14	*4,84E-08*	2,48E-08
			Time(s)	**74,61846**	80,2148	159,9024	167,7352	185,6195	*208,0732*

Table 5: Average results for NMF problem.

6. A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery problems. In D. Palomar and Y.C. Eldar, editors, *Convex Optimization in Signal Processing and Communications*, pages 139–162. Cambridge University Press, Cambridge, 2009.

7. D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 3rd edition, 2016.

8. J. Bolte, S. Sabach, M. Teboulle, and Y. Vaisbourd. First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM J. Optim.*, 28(3):2131–2151, 2018.

9. S. Bonettini, M. Prato, and S Rebegoldi. A new proximal heavy ball inexact line-search algorithm. *Comput. Optim. Appl.*, 88:1–41, 03 2024.

10. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

11. R.E. Bruck. On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in hilbert space. *J. Math. Anal. Appl.*, 61:159–164, 1977.

12. J.Y. Bello Cruz and T.T.A. Nghia. On the convergence of the forward–backward splitting method with linesearches. *Optimization Methods and Software*, 31(6):1209–1238, 2016.

13. R.A. Dragomir, A.B. Taylor, A. d'Aspremont, and J. Bolte. Optimal complexity and certification of bregman first-order methods. *Math. Program.*, 194:41–83, 2022.

14. M.Á.T. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.

15. M. Fukushima and H. Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *Int. J. Syst. Sci.*, 12(8):989–1000, 1981.

16. P.T. Hoai, N.T. Vinh, and N.P.H. Chung. A novel stepsize for gradient descent method. *Operations Research Letters*, page 107072, 2024.

17. X. Jia, C. Kanzow, and P. Mehlitz. Convergence analysis of the proximal gradient method in the presence of the kurdyka–Łojasiewicz property without global lipschitz assumptions. *SIAM J. Optim.*, 33(4):3038–3056, 2023.

18. C. Kanzow and P. Mehlitz. Convergence properties of monotone and nonmonotone proximal gradient methods revisited. *J. Optim. Theory Appl.*, 195(2):624–646, 2022.

19. P. Latafat, A. Themelis, and P. Patrinos. On the convergence of adaptive first order methods: proximal gradient and alternating minimization algorithms. In *Proceedings of Machine Learning Research*, volume 242, pages 197–208, 2024.

20. P. Latafat, A. Themelis, L. Stella, and P. Patrinos. Adaptive proximal algorithms for convex optimization under local lipschitz continuity of the gradient. *Mathematical Programming*, 2024.

21. Q. Lin, R. Ma, and Y. Xu. Complexity of an inexact proximal-point penalty method for constrained smooth non-convex optimization. *Comput. Optim. Appl.*, 82(1):175–224, 2022.

22. H. Liu, T. Wang, and Z. Liu. Some modified fast iterative shrinkage-thresholding algorithms with a new adaptive non-monotone step-size strategy for nonsmooth and convex minimization problems. *Comput. Optim. Appl.*, 83:651–691, 2022.

23. Y. Malitsky and K. Mishchenko. Adaptive gradient descent without descent. In *ICML*, volume 119, pages 6702–6712, 2020.

24. Yura Malitsky and Konstantin Mishchenko. Adaptive proximal gradient method for convex optimization. In *NeurIPS*, 2024.

25. A. De Marchi and A. Themelis. Proximal gradient algorithms under local lipschitz gradient continuity. *J. Optim. Theory Appl.*, 194:771–794, 2022.

26. G.B. Passty. Ergodic convergence to a zero of the sum of monotone operators in hilbert space. *J. Math. Anal. Appl.*, 72:383–390, 1979.

27. P. Symeonidis and A. Zioupos. *Matrix and Tensor Factorization Techniques for Recommender Systems*. Springer Briefs in Computer Science. 2016.

28. M. Teboulle. A simplified view of first order methods for optimization. *Math. Program.*, 170(1):67–96, 2018.

29. A. Themelis, L. Stella, and P. Patrinos. Forward-backward envelope for the sum of two nonconvex functions: further properties and nonmonotone linesearch algorithms. *SIAM J. Optim.*, 28(3):2274–2303, 2018.

30. S.J. Wright, R.D. Nowak, and M.A.T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7):2479–2493, 2009.

31. X. Zhao, R. Raushan, D. Ghosh, JC. Jao, and M. Qi. Proximal gradient method for convex multiobjective optimization problems without lipschitz continuous gradients. *Comput. Optim. Appl.*, 91:27–66, 2025.