Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Adaptive Multi-step Inertial Gradient Method with Line Search for Nonconvex Optimization

Xiaoquan Wang
China University of Mining and Technology
Hu Shao
shaohu@cumt.edu.cn
China University of Mining and Technology

Research Article

Keywords: Nonconvex optimization, Multi-step inertial method, Gradient method, Line search, Complexity analysis

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11438942/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Adaptive Multi-step Inertial Gradient Method with Line Search for Nonconvex Optimization

Xiaoquan Wang[1] and Hu Shao[1*]

[1]School of Mathematics, China University of Mining and Technology, Xuzhou, 221116, Jiangsu, China.

*Corresponding author(s). E-mail(s): shaohu@cumt.edu.cn;
Contributing authors: wxq4869@163.com;

Abstract

In this paper, we consider adaptive inertial first-order methods for nonconvex optimization in smooth and composite settings. For smooth objectives, we propose an adaptive multi-step inertial gradient method with line search. For composite objectives, we extend the scheme to an adaptive multi-step inertial proximal gradient method. The above two algorithms combine the local curvature information of the smooth part of the objective function, the information at stationary points, and the coefficients of multi-step inertia to adaptively control the inertial term. They also employ line search techniques to prevent overshooting caused by the inertial term and to select an appropriate step size, thereby improving the overall performance of the algorithm. Under standard assumptions, we prove finite termination of the line search, global convergence to stationarity, and an $\boldsymbol{O(\epsilon^{-2})}$ complexity bound in terms of the gradient norm or proximal gradient mapping. Numerical results on logistic regression and sparse learning problems are reported

Keywords: Nonconvex optimization, Multi-step inertial method, Gradient method, Line search, Complexity analysis

1 Introduction

Nonconvex optimization has become one of the central topics in modern optimization theory and its applications. A wide range of problems arising from signal processing, machine learning, statistics, image reconstruction, sparse recovery, matrix completion, and deep learning can be formulated as nonconvex optimization models. Typical

examples include sparse regression with nonconvex penalties, low-rank matrix recovery, robust principal component analysis, compressed sensing, dictionary learning, and nonconvex regularized empirical risk minimization [6–9, 12]. Compared with convex models, nonconvex formulations are often able to better capture intrinsic structures of practical problems, such as sparsity, low rankness, discontinuity, and sharp transitions. However, the lack of global convexity also brings substantial challenges to the design and convergence analysis of efficient numerical algorithms.

In this paper, we consider a class of nonconvex optimization problems as follows,

$$\min_{x\in\mathbb{R}^m} F(x). \tag{1}$$

In this paper, we consider two representative $F := f$, where F is possibly nonconvex and smooth and $F := f+h$, where f is possibly nonconvex and smooth and h is proper lower semicontinuous function. The model (1) with above two forms covers many important regularized optimization problems, including ℓ_1-regularized least squares, total variation regularization, constrained optimization through indicator functions, nuclear norm regularization, and sparse or low-rank learning models [4, 5, 10, 13].

When $F := f$, we focus on the nonconvex smooth optimization,

$$\min_{x\in\mathbb{R}^m} f(x). \tag{2}$$

The gradient-type methods are among the most fundamental approaches for solving smooth optimization problems. For problem (2), the classical gradient descent method generates

$$x^{k+1} = x^k - \gamma_k \nabla f(x^k),$$

where $\gamma_k > 0$ is a stepsize. Under the Lipschitz continuity of ∇f, gradient descent with a sufficiently small fixed stepsize admits a standard descent property and converges to a stationary point in the sense that $\|\nabla f(x^k)\| \to 0$ [15]. Nevertheless, the performance of gradient descent is often sensitive to the choice of the stepsize. If the stepsize is too small, the method may converge slowly; if it is too large, the objective value may increase or the algorithm may even diverge. This issue becomes more serious in nonconvex optimization, where the local curvature may vary dramatically along the iterations.

To improve practical performance, adaptive stepsize strategies have been extensively studied. Classical line-search techniques, such as Armijo backtracking, guarantee sufficient descent without requiring an exact knowledge of the global Lipschitz constant. Such strategies are particularly attractive in nonconvex settings because they provide a safeguard against instability caused by local nonconvex curvature. On the other hand, Barzilai-Borwein-type spectral stepsizes and local curvature estimators have also been widely used to accelerate gradient-type methods by exploiting local second-order information implicitly. These adaptive strategies often lead to significantly better numerical performance than conservative fixed stepsize rules, especially for large-scale problems where computing or estimating the global Lipschitz constant is expensive.

Another important technique for accelerating first-order methods is inertia or extrapolation which was inspired by the heavy-ball method [17, 18] or the Nesterov method [14]. The inertial methods incorporate information from previous iterates into the current update. A typical inertial gradient step in [17, 18] takes the form

$$x^{k+1} = x^k - \gamma_k \nabla f(x^k) + \beta_k (x^k - x^{k-1}),$$

where β_k is an inertial parameter. Inertial mechanisms have been shown to be highly effective in both convex and nonconvex optimization. They can reduce oscillations, exploit the historical trajectory, and accelerate convergence in many applications [15, 18]. As suggested by Polyak in [18] , the multi-step inertial techniques, which use several previous displacements rather than only the most recent one, have attracted increasing attention in recent years. Compared with one-step inertia, multi-step inertia can utilize richer historical information and may provide stronger acceleration effects. However, inertial terms may also destroy descent, especially in nonconvex optimization. Therefore, a major theoretical and practical issue is how to design inertial terms that are aggressive enough to accelerate the algorithm but sufficiently controlled to maintain convergence.

When $F := f + h$, which corresponding optimization problem is also called composite optimization,

$$\min_{x \in \mathbb{R}^m} f(x) + h(x). \tag{3}$$

For the problem (3), the proximal gradient methods constitute a standard and powerful class of algorithms. Given a stepsize $\gamma > 0$, the proximal gradient step is

$$x^{k+1} = \mathrm{prox}_{\gamma h}\big(x^k - \gamma \nabla f(x^k)\big),$$

where

$$\mathrm{prox}_{\gamma h}(v) := \arg\min_x \left\{ h(x) + \frac{1}{2\gamma}\|x - v\|^2 \right\},$$

is the proximal operator. The proximal gradient methods and their accelerated variants have been widely investigated for convex and nonconvex composite optimization [1–3] and the inertial strategy has also good performance in proximal gradient methods [11, 16, 21–23]. In particular, the framework of proximal alternating linearized minimization and the Kurdyka-Lojasiewicz property has provided a powerful tool for establishing global convergence of nonconvex nonsmooth algorithms [1, 3, 20].

Despite the success of existing methods, several limitations remain. First, many inertial methods require restrictive conditions on inertial parameters to guarantee convergence. These conditions are often conservative and may limit the acceleration effect in practice. Second, fixed or globally estimated stepsizes may be inefficient for problems with highly nonuniform local curvature. Third, in nonconvex optimization, local curvature estimators may be unreliable and may lead to nonmonotone or unstable behavior if no safeguard is imposed. Fourth, for composite nonsmooth problems, the interaction between the proximal operator, the inertial perturbation, and the sufficient descent condition is more delicate than in the smooth case. Therefore, it is desirable

to develop algorithms that can simultaneously exploit local curvature information, multi-step historical information, and line-search safeguards.

Motivated by these considerations, this paper proposes two adaptive multi-step inertial proximal-gradient-type algorithms for nonconvex optimization. For the smooth problem (2), we propose a nonconvex **a**daptive **m**ulti-step **i**nertial **g**radient method with sufficient **l**ine **s**earch, referred to as AMIGls. At each iteration, the method constructs a multi-step inertial term from a weighted combination of historical displacements. The weights decay geometrically, so that recent directions play a more important role while older directions are still partially retained. The inertial magnitude is adaptively controlled by the current gradient norm, a local curvature estimate, and the historical displacement norm. Furthermore, the inertial perturbation is scaled in the line-search procedure to ensure that it is dominated by the gradient descent component. This design allows the method to enjoy acceleration from historical information while preserving a provable descent property. For the composite problem (3), we generalize AMIGls and propose a nonconvex **a**daptive **m**ulti-step **i**nertial **P**roximal **g**radient **m**ethod with sufficient **l**ine **s**earch, referred to as AMIPGMls. In this method, the trial point is generated by

$$x^+ = \mathrm{prox}_{\gamma^k h}\big(x^k - \gamma^k \nabla f(x^k) + \widetilde{\xi}_k\big),$$

where $\widetilde{\xi}_k$ is a controlled multi-step inertial perturbation. By explicitly enforcing sufficient descent through backtracking, the proposed method remains robust even when the local curvature estimate is inaccurate.

The proposed algorithms have several attractive advantages. First, they incorporate multi-step inertial information instead of using only one previous displacement. This enables the algorithms to better exploit the historical trajectory and may improve practical acceleration. Second, the inertial term is adaptively scaled according to the current stationarity measure. Therefore, the inertial effect naturally weakens as the iterates approach stationarity, which is beneficial for convergence. Third, a local curvature estimator is used to adapt the initial trial stepsize and the inertial magnitude. This avoids the need for a precise global Lipschitz constant and makes the method more responsive to local geometry. Fourth, the sufficient descent line search provides an effective safeguard for nonconvex problems. Even if the curvature estimate is inaccurate or the inertial term is too aggressive, the backtracking strategy can reduce the stepsize until descent is achieved. Finally, the algorithms are simple to implement and require only gradient evaluations and proximal operations, making them suitable for large-scale applications.

From the theoretical perspective, we establish global convergence properties of the proposed methods under standard assumptions. For the smooth case, we prove that the line search terminates finitely at every iteration and that the accepted stepsizes are uniformly bounded away from zero. Based on the sufficient descent condition, we show that the method achieves the standard first-order complexity bound with $O(\epsilon^{-2})$. For the composite case, under finite termination and a positive lower bound on the accepted stepsizes, we prove that the method also has the first-order complexity bound with $O(\epsilon^{-2})$.

The rest of this paper is organized as follows. Section 2 introduces notation and preliminary results used throughout the paper. Section 3 presents the proposed nonconvex AMIGls method for smooth optimization and establishes its descent and convergence properties. Section 4 develops the AMIPGMls method for nonconvex composite optimization and analyzes its proximal gradient complexity. Numerical experiments are reported in Section 5 to demonstrate the efficiency and robustness of the proposed algorithms. Finally, Section 6 concludes the paper.

2 Notations and Preliminaries

In this section, we summarize some notations and preliminaries. For a subset $S \subseteq \mathbb{R}^q$ and a point $x \in \mathbb{R}^q$, if $S \neq \emptyset$, let $\mathrm{dist}(x, S) = \inf_{y \in S} \|y - x\|$, and if $S = \emptyset$, let $\mathrm{dist}(x, S) = +\infty$. For function $f : \mathbb{R}^q \to \mathbb{R} \cup \{+\infty\}$, the effective domain is defined by $\mathrm{dom} f = \{x \in \mathbb{R}^q : f(x) < +\infty\}$. The function f is proper if $\mathrm{dom} f \neq \emptyset$ and $f > -\infty$.

Definition 2.1 [19] Let h: $\mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semi-continuous function.
(i) The Fréchet subdifferential of g at $x \in \mathrm{dom} g$, denoted by $\hat{\partial} h(x)$, is defined as

$$\hat{\partial} h(x) = \left\{ x^* \in \mathbb{R}^m : \lim_{y \neq x} \inf_{y \to x} \frac{h(y) - h(x) - \langle x^*, y - x \rangle}{\|y - x\|} \geqslant 0 \right\}.$$

When $x \notin \mathrm{dom} h$, let $\hat{\partial} h(x) = \emptyset$.
(ii) The limiting-subdifferential of h at $x \in \mathrm{dom} h$ is defined as

$$\partial h(x) = \{x^* \in \mathbb{R}^m : \exists\, x_k \to x, h(x_k) \to h(x), \hat{x}_k \in \hat{\partial} h(x_k), \hat{x}_k \to x^*\}. \tag{4}$$

3 AMIGls and Convergence Analysis

We first give some assumptions for the problem (2).

Assumption 3.1 *The function $f : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and bounded from below. Moreover, ∇f is Lipschitz continuous with constant $L > 0$, namely*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^m.$$

Let $f_{\inf} := \inf_{x \in \mathbb{R}^m} f(x) > -\infty$.

Then we give detail explanations and framework of AMIGls. It is noteworthy that the classical gradient-type methods rely on a globally conservative Lipschitz constant are often robust but conservative, thus we would like to seek to exploit local geometry instead. We have following definition.

Definition 3.1 *For any $u, v \in \mathbb{R}^m$, define the local curvature estimator*

$$L(u, v) := -\frac{1}{2} \frac{\|\nabla f(u) - \nabla f(v)\|^2}{f(u) - f(v) + \langle \nabla f(u), v - u \rangle}, \tag{5}$$

whenever the denominator is nonzero. In the nonconvex case, the estimator is truncated as

$$\widehat{L}_{k+1} = \min\left\{ L_{\max}, \max\left\{ L_{\min}, |L(x^k, x^{k-1})| \right\} \right\}, \tag{6}$$

where $0 < L_{\min} \leq L_{\max} < \infty$.

The quantity $\widehat{L}_k$ is a truncated local curvature estimator. Its purpose is to approximate the local smoothness level of f along the current trajectory without requiring knowledge of the global Lipschitz constant L. In practice, this makes the method more adaptive and often less conservative than fixed-step gradient descent. The truncation between $L_{\min}$ and $L_{\max}$ prevents the estimator from becoming numerically unstable, excessively small, or overly large.

We now consider the inertial strategy. The AMIGls starts from an initial point x^0 and extends it artificially to $x^{-m} = \cdots = x^{-1} = x^0$. This initialization is standard in multi-step inertial schemes and ensures that the first few historical displacements are well-defined. The memory length $m \geq 1$ determines how many previous steps are used to form the inertial term. A larger m allows the method to capture richer trajectory information, but also increases the risk of over-extrapolation. The geometric weights q^{j-1} with $q \in (0, 1)$ naturally prioritize recent displacements while still retaining a fading memory of older ones. We also know that the inertial strategy could accelerate the algorithm but may lose stability if the extrapolation is too aggressive. To take advantages of the inertial strategy, we try to balance these two aspects by letting the inertial strength depend on the current stationarity level and by scaling the extrapolation with the local curvature estimate. As a result, the algorithm becomes more aggressive in the early stage and automatically becomes more cautious near a stationary point. Then we enforce a sufficient-descent condition through backtracking so that the inertial term and the adaptive stepsize remain under control even in the presence of nonconvexity. We denote $d_k^{(j)} = x^{k-j+1} - x^{k-j}, \quad j = 1, \ldots, m$, and introduce the term

$$H_k = \left(\sum_{j=1}^{m} q^{2(j-1)} \|d_k^{(j)}\|^2 \right)^{1/2}, \tag{7}$$

to measure the magnitude of recent motion of the iterates. It plays the role of a local history indicator: if the iterates are moving rapidly, then H_k becomes large and the algorithm reduces the inertial magnitude, if the iterates are stabilizing, then H_k becomes small and the algorithm may permit a larger extrapolation. This mechanism helps the method adapt automatically to different phases of the optimization process. Then the inertial parameter is chosen as

$$\omega_k = \min \left\{ \omega_{\max}, \eta \frac{\|\nabla f(x^k)\|}{\widehat{L}_k H_k + \varepsilon} \right\}. \tag{8}$$

This formula reflects the central idea of the method. The factor $\|\nabla f(x^k)\|$ measures the current nonstationarity, so the inertial effect is stronger when the gradient is large and weaker near stationarity. The denominator combines local curvature information and historical displacement size, thereby preventing excessive extrapolation when the local geometry is steep or the trajectory is highly oscillatory. The cap $\omega_{\max}$ is introduced only for safety and does not affect the basic structure of the method. The raw inertial vector

$$\widetilde{\xi}_k = \sum_{j=1}^{m} \omega_k q^{j-1} d_k^{(j)} \tag{9}$$

aggregates several past displacements. Compared with one-step inertia, this construction can better exploit the curvature and momentum accumulated over several iterations. The geometric decay q^{j-1} ensures that earlier displacements have progressively smaller influence. This is important because older displacements may correspond to outdated local geometry and should not dominate the current update.

We give the initial trial stepsize of gradient method, which is set to

$$\gamma^k = \min\left\{\gamma_{\max}, \frac{c}{\widehat{L}_k}\right\}. \tag{10}$$

This is a curvature-aware initialization. If the estimated curvature is high, the trial stepsize is small; if the estimated curvature is mild, the method allows a larger trial step. This choice significantly reduces the number of backtracking reductions in practice. On this basis, we need to employ a line search technique to determine an appropriate step size γ^k. However, as discussed earlier, the inertial step may disrupt the structure of the line search. Therefore, it is necessary to impose a restriction (11) on the inertial term, the role of which will be introduced later.

$$\alpha_k(\gamma^k) = \min\left\{1, \frac{\theta\gamma^k\|\nabla f(x^k)\|}{\|\widetilde{\xi}_k\| + \varepsilon}\right\}, \quad \widetilde{\xi}_k = \alpha_k(\gamma^k)\xi_k. \tag{11}$$

The trial point is

$$x^+ = x^k - \gamma^k\nabla f(x^k) + \widetilde{\xi}_k.$$

This is a gradient step perturbed by a controlled inertial correction. The acceptance criterion

$$f(x^+) \le f(x^k) - \sigma\gamma^k\|\nabla f(x^k)\|^2,$$

forces a strict decrease in the objective. This condition is stronger than mere nonincrease and is essential for convergence analysis in the nonconvex setting. It prevents the algorithm from accepting excessively long or unstable steps. Moreover, it is worth emphasizing that when $q = 0$, the algorithm loses its inertial effect and reduces to a class of gradient methods that perform line searches using only local curvature information. We now give the detailed AMIGls in Algorithm 1.

Next, we present the convergence analysis of AMIGls. We first show that the line search in AMIGls is well defined.

Lemma 3.1 *For Algorithm 1, for every trial stepsize $\gamma^k > 0$, the scaled inertial term satisfies*

$$\|\widetilde{\xi}_k\| \le \theta\gamma^k\|\nabla f(x^k)\|. \tag{12}$$

Proof By the definition of $\alpha_k(\gamma^k)$,

$$\alpha_k(\gamma^k) \le \frac{\theta\gamma^k\|\nabla f(x^k)\|}{\|\xi_k\| + \varepsilon}.$$

Hence

$$\|\widetilde{\xi}_k\| = \alpha_k(\gamma^k)\|\xi_k\| \le \frac{\theta\gamma^k\|\nabla f(x^k)\|\|\xi_k\|}{\|\xi_k\| + \varepsilon} \le \theta\gamma^k\|\nabla f(x^k)\|.$$

Algorithm 1 AMIGls

Choose $x^0 \in \mathbb{R}^d$, memory length $m \geq 1$, $q \in [0,1)$, $\eta > 0$, $\omega_{\max} > 0$, $\theta \in [0,1)$, $\sigma \in (0, 1-\theta)$, $\tau > 1$, $c > 0$, $s_{\max} > 0$, $0 < L_{\min} \leq L_{\max} < \infty$, and $\varepsilon > 0$.
Set $x^{-m} = \cdots = x^{-1} = x^0$ and choose $\widehat{L}_0 \in [L_{\min}, L_{\max}]$.
for $k = 0, 1, 2, \ldots$ **do**
    Compute historical displacements

$$d_k^{(j)} = x^{k-j+1} - x^{k-j}, \quad j = 1, \ldots, m.$$

    Compute the weighted historical norm as (7).
    Set the inertial parameter as (8).
    Form the raw multi-step inertial term (9).
    Initialize the stepsize with (10).
    repeat
        Compute $\xi_k(s)$ with (11).
        Compute the trial point

$$x^+ = x^k - \gamma^k \nabla f(x^k) + \xi_k(s).$$

        if $f(x^+) \leq f(x^k) - \sigma\gamma^k \|\nabla f(x^k)\|^2$ **then**
            Accept $x^{k+1} = x^+$.
        else
            Set $\gamma^k \leftarrow \gamma^k / \tau$.
        end if
    until the trial point is accepted
    Update $\widehat{L}_{k+1}$ by (6).
end for

□

This is a crucial safeguard. It guarantees that the perturbation is never too large relative to the gradient step. In particular, the bound

$$\|\widetilde{\xi}_k\| \leq \theta\gamma^k \|\nabla f(x^k)\|$$

is precisely what enables the sufficient-descent proof. In other words, the inertial term is allowed to accelerate the method, but it is always subordinate to the gradient descent component.

Lemma 3.2 *Suppose Assumption 3.1 holds. Let $\theta \in [0,1)$ and $\sigma \in (0, 1-\theta)$. Then, at every iteration k, the line search in Algorithm 1 terminates after finitely many reductions. Moreover, if the initial trial stepsizes are bounded above by $\gamma_{\max}$, then the accepted stepsizes satisfy*

$$\gamma^k \geq \hat{\gamma}_{\min} > 0, \tag{13}$$

where one may take

$$\hat{\gamma}_{\min} := \min\left\{\gamma_{\max}, \frac{c}{L_{\max}}, \frac{1}{\tau}\frac{2(1-\theta-\sigma)}{L(1+\theta)^2}\right\}. \tag{14}$$

Proof Let

$$p_k(\gamma^k) = x^+ - x^k = -\gamma^k \nabla f(x^k) + \widetilde{\xi}_k.$$

By Lemma 3.1,

$$\langle \nabla f(x^k), p_k(\gamma^k)\rangle = -\gamma^k \|\nabla f(x^k)\|^2 + \langle \nabla f(x^k), \widetilde{\xi}_k\rangle \le -(1-\theta)\gamma^k \|\nabla f(x^k)\|^2,$$

and

$$\|p_k(\gamma^k)\| \le \gamma^k \|\nabla f(x^k)\| + \|\widetilde{\xi}_k\| \le (1+\theta)\gamma^k \|\nabla f(x^k)\|.$$

By the descent lemma for functions with Lipschitz continuous gradient,

$$\begin{aligned} f(x^+) &\le f(x^k) + \langle \nabla f(x^k), p_k(\gamma^k)\rangle + \frac{L}{2}\|p_k(\gamma^k)\|^2 \\ &\le f(x^k) - \left[(1-\theta)\gamma^k - \frac{L}{2}(1+\theta)^2(\gamma^k)^2\right]\|\nabla f(x^k)\|^2. \end{aligned}$$

Therefore the sufficient descent condition

$$f(x^+) \le f(x^k) - \sigma\gamma^k \|\nabla f(x^k)\|^2$$

holds whenever

$$(1-\theta)\gamma^k - \frac{L}{2}(1+\theta)^2(\gamma^k)^2 \ge \sigma\gamma^k,$$

that is, whenever

$$\gamma^k \le \bar{\gamma} := \frac{2(1-\theta-\sigma)}{L(1+\theta)^2} > 0.$$

Since backtracking replaces γ^k by γ^k/τ, finite termination follows. The lower bound (14) follows from the boundedness of the initial trial stepsize and the fact that the first accepted stepsize is larger than $\bar{\gamma}/\tau$ unless it is accepted before reaching $\bar{\gamma}$. □

Lemma 3.3 *Suppose Assumption 3.1 holds. Then the sequence generated by Algorithm 1 satisfies*

$$f(x^{k+1}) \le f(x^k) - \sigma\gamma^k \|\nabla f(x^k)\|^2, \quad k = 0, 1, 2, \ldots. \tag{15}$$

Consequently, $\{f(x^k)\}$ is monotonically nonincreasing and convergent.

Proof The inequality (15) is exactly the acceptance condition in Algorithm 1. Because f is bounded from below by Assumption 3.1, the monotone sequence $\{f(x^k)\}$ converges. □

Theorem 3.1 *Suppose Assumption 3.1 holds. Let $\{x^k\}$ be generated by Algorithm 1. Then*

$$\sum_{k=0}^{\infty} \|\nabla f(x^k)\|^2 < \infty, \qquad \lim_{k\to\infty} \|\nabla f(x^k)\| = 0. \tag{16}$$

Moreover, for every integer $N \ge 0$,

$$\min_{0\le k\le N} \|\nabla f(x^k)\|^2 \le \frac{f(x^0) - f_{\inf}}{\sigma s_{\min}(N+1)}. \tag{17}$$

Consequently, Algorithm 1 finds an ε-stationary point, namely a point x^k satisfying

$$\|\nabla f(x^k)\| \le \epsilon,$$

in at most $O(\epsilon^{-2})$ iterations.

Proof By the acceptance criterion,

$$f(x^{k+1}) \le f(x^k) - \sigma\gamma^k \|\nabla f(x^k)\|^2.$$

Using Lemma 3.2, $\gamma^k \ge \hat{\gamma}_{\min} > 0$, hence

$$\sigma\hat{\gamma}_{\min}\|\nabla f(x^k)\|^2 \le f(x^k) - f(x^{k+1}).$$

Summing from $k = 0$ to N gives

$$\sigma\hat{\gamma}_{\min} \sum_{k=0}^{N} \|\nabla f(x^k)\|^2 \le f(x^0) - f(x^{N+1}) \le f(x^0) - f_{\inf}.$$

Letting $N \to \infty$ yields the summability in (16). Therefore $\|\nabla f(x^k)\| \to 0$. Also,

$$(N+1) \min_{0\le k\le N} \|\nabla f(x^k)\|^2 \le \sum_{k=0}^{N} \|\nabla f(x^k)\|^2,$$

which gives (17). The $O(\epsilon^{-2})$ bound follows immediately. □

Corollary 3.1 *Suppose the assumptions of Theorem 3.1 hold and the sequence $\{x^k\}$ is bounded. Then every cluster point of $\{x^k\}$ is a stationary point of f.*

Proof Let $\bar{x}$ be a cluster point. Then there exists a subsequence $\{x^{k_j}\}$ such that $x^{k_j} \to \bar{x}$. By Theorem 3.1, $\|\nabla f(x^{k_j})\| \to 0$. Since ∇f is continuous, $\nabla f(\bar{x}) = 0$. □

4 AMIPGMls and Convergence Analysis

In this section, we extend the techniques embedded in AMIGls to the proximal gradient setting, yielding AMIPGMls. We first give some assumptions on the problem (3).

Assumption 4.1 *The following conditions hold.*

1. *$f : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and ∇f is Lipschitz continuous with constant $L > 0$.*
2. *$h : \mathbb{R}^m \to (-\infty, +\infty]$ is proper, lower semicontinuous.*
3. *$F = f + h$ is bounded from below on $\mathbb{R}^m$, and $F_{\inf} := \inf_{x\in\mathbb{R}^m} F(x) > -\infty$.*
4. *For every $\gamma > 0$, the proximal operator*

$$\operatorname{prox}_{\gamma h}(v) = \arg\min_x \left\{ h(x) + \frac{1}{2\gamma}\|x - v\|^2 \right\}$$

is well defined and nonexpansive.

From the perspective of algorithmic structure, this extension is quite natural; however, it should be noted that in Algorithm 1, the inertial coefficient is scaled using $\|\nabla f(x^k)\|$ which is not appropriate for the composite optimization. Then, we denote the quantity

$$G_\gamma(x) = \frac{1}{\gamma}\left(x - \mathrm{prox}_{\gamma h}(x - \gamma \nabla f(x))\right) \tag{18}$$

This is the natural generalization of the gradient to composite problems. If $G_\gamma(x) = 0$, then x satisfies the first-order condition

$$0 \in \nabla f(x) + \partial h(x).$$

Thus, controlling $\|G_{\gamma^k}(x^k)\|$ provides a principled way to measure convergence. In addition, it should be made clear that γ or γ^k here denote the step sizes required for computing the proximal operator, although they are initialized in the same manner as in Algorithm 1. This choice remains natural in the composite setting because the smooth part f is still governed by local curvature. The proximal step then handles the nonsmooth term exactly, which helps preserve stability and often improves robustness relative to linearized nonsmooth treatments.

Subsequently, we compute $\widehat{L}_k$, H_k and ω^k in the same manner as in Algorithm 1. To ensure that the line search proceeds smoothly, we still need to impose a constraint on the total inertia. Unlike in (11), however, the update must be carried out using the quantity $G_\gamma(x)$. We therefore obtain the following inertial constraint:

$$\alpha_k(\gamma^k) = \min\left\{1, \frac{\theta \gamma^k \|G_{\gamma^k}(x^k)\|}{\|\xi_k\| + \varepsilon}\right\}, \qquad \widetilde{\xi}_k = \alpha_k(\gamma^k)\xi_k. \tag{19}$$

Now we can give the AMIPGMls in Algorithm 2.

Algorithm 2 AMIPGMls

Choose $x^0 \in \operatorname{dom} h$, memory length $m \geq 1$, $q \in [0,1)$, $\eta > 0$, $\omega_{\max} > 0$, $\theta \in [0, \sqrt{5}-2)$, $\sigma > 0$, $\tau > 1$, $c > 0$, $s_{\max} > 0$, $0 < L_{\min} \leq L_{\max} < \infty$, and $\varepsilon > 0$.
Set $x^{-m} = \cdots = x^{-1} = x^0$ and choose $\widehat{L}_0 \in [L_{\min}, L_{\max}]$.
for $k = 0, 1, 2, \ldots$ **do**
    Compute
$$d_k^{(j)} = x^{k-j+1} - x^{k-j}, \quad j = 1, \ldots, m.$$
    Compute the weighted historical norm as (7).
    Set the inertial parameter as (8).
    Initialize the stepsize γ^k with (10).
    Compute $G_{\gamma^k}(x^k)$ from (18).
    Form the raw multi-step inertial term (9).
    repeat
        Recompute $G_{\gamma^k}(x^k)$.
        Compute $\xi_k(s)$ with (19).
        Compute the trial point
$$x^+ = \operatorname{prox}_{\gamma^k h}\big(x^k - \gamma^k \nabla f(x^k) + \widetilde{\xi}_k\big).$$
        if $F(x^+) \leq F(x^k) - \sigma\gamma^k \|G_{\gamma^k}(x^k)\|^2$ **then**
            Accept $x^{k+1} = x^+$.
        else
            Set $\gamma^k \leftarrow \gamma^k / \tau$.
        end if
    until the trial point is accepted
    Update $\widehat{L}_{k+1}$ by (6)
end for

Although AMIPGMls can be regarded as a direct generalization of AMIGls, its convergence analysis differs slightly, particularly in the establishment of sufficient descent under the line-search scheme. We first show that the line search in AMIPGMls is well defined.

Lemma 4.1 *For Algorithm 2, for every trial stepsize $\gamma^k > 0$, the scaled inertial term satisfies*
$$\|\widetilde{\xi}_k\| \leq \theta\gamma^k \|G_{\gamma^k}(x^k)\|.$$

Lemma 4.2 *Suppose that Assumption 4.1 holds. Let $\theta \in [0,1)$ and assume that*
$$0 < \sigma < \frac{1 - 4\theta - \theta^2}{2}.$$
Then, at every iteration k, the line search in Algorithm 2 terminates after finitely many reductions. Moreover, if the initial trial stepsizes are bounded above by $s_{\max}$, then the accepted stepsizes satisfy
$$\gamma^k \geq \hat{\gamma}_{\min} > 0,$$

where one may take

$$\hat{\gamma}_{\min} := \min\left\{\hat{\gamma}_{\max}, \frac{c}{L_{\max}}, \frac{1}{\tau}\frac{1-\theta^2-2\sigma}{L(1+\theta)^2}\right\}.$$

Proof Fix an iteration index k. Denote

$$\nabla f(x^k) := \nabla f(x^k), \qquad G := G_{\gamma^k}(x^k).$$

The trial point generated in Algorithm 2 is

$$x^+ = \mathrm{prox}_{\gamma^k h}\big(x^k - \gamma^k \nabla f(x^k) + \tilde{\xi}^k\big).$$

Moreover, define the unperturbed proximal point

$$z := \mathrm{prox}_{\gamma^k h}(x^k - \gamma^k \nabla f(x^k)).$$

By the definition of the proximal gradient mapping,

$$G_{\gamma^k}(x^k) = \frac{1}{\gamma^k}(x^k - z),$$

and hence

$$\|z - x^k\| = \gamma^k \|G_{\gamma^k}(x^k)\|.$$

Since the proximal operator of a proper closed convex function is nonexpansive, we have

$$\|x^+ - z\| \le \|\tilde{\xi}^k\|.$$

Using the scaling condition of the inertial perturbation,

$$\|\tilde{\xi}^k\| \le \theta\gamma^k \|G_\gamma^k(x^k)\|,$$

it follows that

$$\|x^+ - x^k\| \le \|x^+ - z\| + \|z - x^k\| \le \|\tilde{\xi}^k\| + \gamma^k\|G_\gamma^k(x^k)\| \le (1+\theta)\gamma^k\|G_\gamma^k(x^k)\|,$$

and

$$\|x^+ - x^k\| \ge \|z - x^k\| - \|x^+ - z\| \ge \gamma^k\|G_\gamma^k(x^k)\| - \|\tilde{\xi}^k\| \ge (1-\theta)\gamma^k\|G_\gamma^k(x^k)\|.$$

Let

$$\Delta x^+ := x^+ - x^k.$$

By the definition of x^+ as a proximal point, for any x, we have

$$h(x^+) + \frac{1}{2\gamma^k}\|x^+ - (x^k - \gamma^k\nabla f(x^k) + \tilde{\xi}^k)\|^2 \le h(x) + \frac{1}{2\gamma^k}\|x - (x^k - \gamma^k\nabla f(x^k) + \tilde{\xi}^k)\|^2.$$

Taking $x = x^k$ gives

$$h(x^+) + \frac{1}{2\gamma^k}\|x^+ - x^k + \gamma^k\nabla f(x^k) - \tilde{\xi}^k\|^2 \le h(x^k) + \frac{1}{2\gamma^k}\|\gamma^k\nabla f(x^k) - \tilde{\xi}^k\|^2.$$

Since $\Delta x^+ = x^+ - x^k$, this becomes

$$h(x^+) + \frac{1}{2\gamma^k}\|\Delta x^+ + \gamma^k\nabla f(x^k) - \tilde{\xi}^k\|^2 \le h(x^k) + \frac{1}{2\gamma^k}\|\gamma^k\nabla f(x^k) - \tilde{\xi}^k\|^2.$$

Expanding the squared norm yields

$$h(x^+) \le h(x^k) - \langle\nabla f(x^k), \Delta x^+\rangle + \frac{1}{\gamma^k}\langle\Delta x^+, \tilde{\xi}^k\rangle - \frac{1}{2\gamma^k}\|\Delta x^+\|^2.$$

On the other hand, since ∇f is Lipschitz continuous with constant L, the descent lemma gives

$$f(x^+) \le f(x^k) + \langle \nabla f(x^k), \Delta x^+ \rangle + \frac{L}{2}\|\Delta x^+\|^2.$$

Adding the last two inequalities, we obtain

$$F(x^+) \le F(x^k) + \left(\frac{L}{2} - \frac{1}{2\gamma^k}\right)\|\Delta x^+\|^2 + \frac{1}{\gamma^k}\langle \Delta x^+, \tilde{\xi}^k \rangle.$$

Using the Cauchy–Schwarz inequality, together with

$$\|\Delta x^+\| \le (1+\theta)\gamma^k \|G_{\gamma^k}(x^k)\|, \qquad \|\tilde{\xi}^k\| \le \theta\gamma^k \|G_{\gamma^k}(x^k)\|,$$

we have

$$\frac{1}{\gamma^k}\langle \Delta x^+, \tilde{\xi}^k \rangle \le \frac{1}{\gamma^k}\|\Delta x^+\|\|\tilde{\xi}^k\| \le \theta(1+\theta)\gamma^k \|G_{\gamma^k}(x^k)\|^2,$$

and

$$\|\Delta x^+\|^2 \le (1+\theta)^2 {\gamma^k}^2 \|G_{\gamma^k}(x^k)\|^2.$$

Therefore, when $\gamma^k \le \frac{1}{L}$, such that

$$\begin{aligned} F(x^+) &\le F(x^k) + \left(\frac{L}{2} - \frac{1}{2\gamma^k}\right)(1-\theta)^2 {\gamma^k}^2 \|G_{\gamma^k}(x^k)\|^2 + \theta(1+\theta)\gamma^k \|G_{\gamma^k}(x^k)\|^2 \\ &= F(x^k) - \gamma^k \left[\frac{(1-\theta)^2}{2} - \theta(1+\theta) - \frac{L}{2}(1-\theta)^2\gamma^k\right]\|G_{\gamma^k}(x^k)\|^2 \\ &= F(x^k) - \gamma^k \left[\frac{1-4\theta-\theta^2}{2} - \frac{L}{2}(1-\theta)^2\gamma^k\right]\|G_{\gamma^k}(x^k)\|^2. \end{aligned}$$

Hence, if

$$\gamma^k \le \bar{\gamma} := \frac{1-4\theta-\theta^2-2\sigma}{L(1-\theta)^2},$$

then $\bar{\gamma} > 0$ and

$$\frac{1-4\theta-\theta^2}{2} - \frac{L}{2}(1-\theta)^2\gamma^k \ge \sigma.$$

Consequently,

$$F(x^+) \le F(x^k) - \sigma\gamma^k \|G_{\gamma^k}(x^k)\|^2,$$

which is exactly the sufficient descent condition in Algorithm 2.

Since the backtracking line search replaces γ^k by γ^k/τ with $\tau > 1$ whenever the sufficient descent condition fails, the trial stepsizes converge to zero after repeated reductions. Therefore, after finitely many reductions, one must have $\gamma^k \le \bar{\gamma}$, and the sufficient descent condition is satisfied. Thus, the line search terminates after finitely many reductions. □

Lemma 4.3 *Suppose Assumption 4.1 holds. Then the sequence generated by Algorithm 2 satisfies*

$$F(x^{k+1}) \le F(x^k) - \sigma\gamma^k \|G_{\gamma^k}(x^k)\|^2, \quad k = 0, 1, 2, \ldots. \tag{20}$$

Consequently, $\{F(x^k)\}$ is monotonically nonincreasing and convergent.

Proof The inequality (20) is exactly the acceptance condition in Algorithm 2. Since $\gamma^k > 0$ and $\sigma > 0$, it follows that

$$F(x^{k+1}) \le F(x^k).$$

Because F is bounded from below by Assumption 4.1, the monotone sequence $\{F(x^k)\}$ converges. □

Theorem 4.1 *Suppose Assumption 4.1 holds. Let $\{x^k\}$ be generated by Algorithm 2. Then*

$$\sum_{k=0}^{\infty} \|G_{\gamma^k}(x^k)\|^2 < \infty, \qquad \lim_{k\to\infty} \|G_{\gamma^k}(x^k)\| = 0. \tag{21}$$

Moreover, for every integer $N \geq 0$,

$$\min_{0\leq k\leq N} \|G_{\gamma^k}(x^k)\|^2 \leq \frac{F(x^0) - F_{\inf}}{\sigma\hat{\gamma}_{\min}(N+1)}. \tag{22}$$

Consequently, Algorithm 2 finds an ε-proximal stationary point, namely a point x^k satisfying

$$\|G_{\gamma^k}(x^k)\| \leq \epsilon,$$

in at most $O(\varepsilon^{-2})$ iterations.

Proof By Lemmas 4.2 and 4.3,

$$\sigma\hat{\gamma}_{\min}\|G_{\gamma^k}(x^k)\|^2 \leq F(x^k) - F(x^{k+1}).$$

Summing from $k = 0$ to N gives

$$\sigma\hat{\gamma}_{\min} \sum_{k=0}^{N} \|G_{\gamma^k}(x^k)\|^2 \leq F(x^0) - F(x^{N+1}) \leq F(x^0) - F_{\inf}.$$

Letting $N \to \infty$ yields

$$\sum_{k=0}^{\infty} \|G_{\gamma^k}(x^k)\|^2 < \infty,$$

and hence $\|G_{\gamma^k}(x^k)\| \to 0$. Furthermore,

$$(N+1) \min_{0\leq k\leq N} \|G_{\gamma^k}(x^k)\|^2 \leq \sum_{k=0}^{N} \|G_{\gamma^k}(x^k)\|^2,$$

which proves (21) and (22). The $O(\epsilon^{-2})$ iteration bound follows immediately. □

Lemma 4.4 *Suppose Assumption 4.1. Assume further that $\{x^k\}$ is bounded and that the stepsizes satisfy $\gamma_k \leq \gamma_{\max} < \infty$. If $\bar{x}$ is a cluster point of $\{x^k\}$ and the proximal gradient mapping is continuous along the convergent subsequence, then*

$$0 \in \nabla f(\bar{x}) + \partial h(\bar{x}). \tag{23}$$

Proof Let $x^{k_j} \to \bar{x}$. By Theorem 4.1,

$$\|G_{\gamma^{k_j}}(x^{k_j})\| \to 0.$$

Let

$$y^{k_j} = \operatorname{prox}_{\gamma^{k_j} h}\big(x^{k_j} - \gamma^{k_j}\nabla f(x^{k_j})\big).$$

Then

$$G_{\gamma^{k_j}}(x^{k_j}) = \frac{x^{k_j} - y^{k_j}}{\gamma^{k_j}},$$

and since $\gamma^{k_j} \leq \gamma_{\max}$,

$$\|x^{k_j} - y^{k_j}\| \leq \gamma_{\max}\|G_{\gamma^{k_j}}(x^{k_j})\| \to 0.$$

Thus $y^{k_j} \to \bar{x}$. By the optimality condition of the proximal step,

$$0 \in \partial h(y^{k_j}) + \nabla f(x^{k_j}) + \frac{1}{\gamma^{k_j}}(y^{k_j} - x^{k_j}).$$

Equivalently,

$$-\nabla f(x^{k_j}) + G_{\gamma^{k_j}}(x^{k_j}) \in \partial h(y^{k_j}).$$

Passing to the limit and using the continuity of ∇f, the convergence of $G_{\gamma^{k_j}}(x^{k_j})$ to zero, and the closedness of the graph of ∂h for function h, we obtain

$$-\nabla f(\bar{x}) \in \partial h(\bar{x}),$$

which obtains (23). □

5 Numerical Experiments

In this section, we conduct numerical experiments to evaluate the performance of the proposed AMIGls method for ℓ_2-regularized logistic regression and AMIPGMls method for SCAD penalty problem to verify the performance of the proposed methods. All numerical experiments are carried out in 64-bit MATLAB R2019b on 64-bit PC with Intel(R) core(TM)i7-6700HQ CPU@2.6GHz and 32GB of RAM.

5.1 ℓ_2-regularized logistic regression

We test the proposed method on a large-scale binary classification problem with an ℓ_2-regularized logistic regression model:

$$\min f(x) := \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-b_i a_i^\top x)\right) + \frac{\lambda}{2}\|x\|^2, \tag{28}$$

where $a_i \in \mathbb{R}^m$ is the feature vector, $b_i \in \{-1, 1\}$ is the class label and $x \in \mathbb{R}^m$ is the weight vector. For this problem, it is easy to know that the f is L-smooth with $L = \frac{1}{4n}\|A\|^2 + \lambda$, where $A = (a_1^\top, \ldots, a_n^\top)^\top$.

To illustrate the performance of AMIGls, We compare the proposed AMIGls method with the following baselines:

- AMIGls-I with non-inertial accelerated ($m = 0$ in Algorithm 1);
- AMIGls-II with 2-step inertial accelerated ($m = 2$ in Algorithm 1);
- Gradient descent with backtracking line search (GD);
- Nesterov accelerated gradient (NAG) in [14].

For the proposed AMIGls method, we use the parameter setting recommended by the implementation: $q = 0.6$, $\eta = 0.8$, $\omega_{\max} = 0.95$, $\theta = 0.22$, $\sigma = 10^{-4}$, $\tau = 2$, $c = 1$, $\gamma_{\max} = \frac{10}{\max\{L, 10^{-12}\}}$, $L_{\min} = 10^{-12}$, $\varepsilon = 10^{-12}$, where L_0 is an initial Lipschitz estimate computed from the augmented data matrix. For GD, the same sufficient descent parameter and backtracking ratio are used. For NAG, the initial stepsize is set according to the same Lipschitz estimate.

In this experiment, the regularization parameter is set to $\lambda = 10^{-3}$, and the initial point is chosen as $x^0 = 0$. we run all algorithms for following termination condition

$$\|\nabla f(x^k)\| \le 10^{-8} \quad \text{or} \quad k > 1500.$$

$(n = 2000, m = 500)$	Iter	Obj	$\log(\|\nabla f(x^k)\|)$	Time (s)
AMIGls-I	335	0.0537	-18.4480	0.0360
AMIGls-II	232	0.0537	-18.4380	0.0213
GD	1500	0.0537	-12.2820	0.1260
NAG	1500	0.0537	-15.7240	0.2401
$(n = 3000, m = 1000)$				
AMIGls-I	488	0.0466	-18.4240	0.1303
AMIGls-II	342	0.0466	-18.4410	0.0722
GD	1500	0.0466	-9.8936	0.2769
NAG	1500	0.0466	-14.4380	0.5307
$(n = 4000, m = 2000)$				
AMIGls-I	740	0.0332	-18.4240	0.6558
AMIGls-II	522	0.0332	-18.4410	0.4234
GD	1500	0.0332	-9.8936	1.2045
NAG	1500	0.0332	-14.4380	2.2766

Table 1: Experimental results with different (n, d)

In Table 1, We report results with following performance metrics: iteration numbers (Iter), final objective value (Obj), $\log(\nabla f(x^k))$ and total running time in seconds (Time (s)).

To compare the convergence behavior more clearly, we also plot the objective gap to the best final value among all methods (denote $\log(|f(x^k) - f^*|)$) along with iteration numbers in Figure 1.

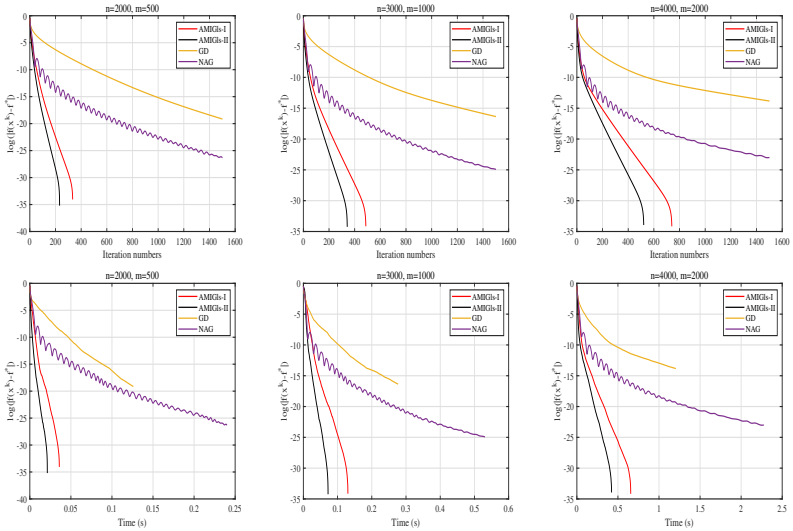


Fig. 1: Comparison of different algorithms

From the above experimental results, we can draw the following conclusions. Compared with NAG and GD, AMIGls exhibits favorable numerical performance even without inertia, that is, under the zero-inertia setting. It requires fewer iterations and less computational time, indicating that the line search strategy and the use of local curvature in place of the global gradient Lipschitz constant can effectively improve both the stability and efficiency of the algorithm. After incorporating a single inertial step, AMIGls shows further performance gains, with substantial reductions in computational time and iteration count, which also demonstrates the effectiveness of the inertial strategy.

5.2 SCAD penalty problem

In this subsection, we apply the proposed method to solve the SCAD penalty problem in statistical learning. The SCAD penalty problem [8] can be characterized as

$$\min_{x} \sum_{i=1}^{n} g_\kappa(\mid x_i \mid) + \frac{1}{2}\|Ax-b\|^2 \tag{24}$$

where $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$, and $g_\kappa(\cdot)$ is defined by

$$g_\kappa(\theta) = \begin{cases} \kappa\theta, & \theta \leqslant \kappa, \\ \frac{-\theta^2+2c\kappa\theta-\kappa^2}{2(c-1)}, & \kappa<\theta \leqslant c\kappa, \\ \frac{(c+1)\kappa^2}{2}, & \theta>c\kappa, \end{cases}$$

with $c > 2$ and $\kappa > 0$ being the knots of quadratic spline function. Noticing that the solution of the following problem

$$\min_{x\in\mathbb{R}^m} \frac{1}{2v}\|x-q\|^2 + \sum_{i=1}^{n} g_\kappa(\mid x_i \mid),$$

with $1+v \leqslant c$ and known q, can be produced by [8, 24], i.e.,

$$x_i := \begin{cases} \mathrm{sign}(q_i)(\mid q_i \mid -\kappa v)_+, & \mid q_i \mid \leqslant (1+v)\kappa, \\ \frac{(c-1)q_i-\mathrm{sign}(q_i)c\kappa v}{c-1-v}, & (1+v)\kappa< \mid q_i \mid \leqslant c\kappa, \\ q_i, & \mid q_i \mid >c\kappa. \end{cases}$$

We can transform (24) into the sum of a smooth nonconvex function f and a convex nonsmooth function h, where

$$f(x) = \frac{1}{2}\|Ax-b\|^2 - \frac{1}{2}\|x\|^2,\ h(x) = \sum_{i=1}^{n} g_\kappa(\mid x_i \mid) + \frac{1}{2}\|x\|^2.$$

It is easy to see that f has Lipschitz continuous gradient and h is convex [24]. Then, we could apply the AMIPGMls to the problem (24) due to $\mathrm{prox}_{\gamma h}(\cdot)$ is nonexpansive.

In our experiments, we choose $A_{ij} \sim \mathcal{N}(0,1)$ and then normalize the columns. Next, we generate a random spares vector in $\mathbb{R}^n$ with the density $\frac{100}{n}$ as x^*, and the vector $b = Ax^* + \epsilon$ with the noise vector $\epsilon \sim \mathcal{N}(0, 0.01I)$. For the sole purpose of showing the numerical efficiency, we simply take some parameters as constants, i.e., $c = 3.7$, $\kappa = 0.1$. To illustrate the performance of AMIPGMls, We conducted comparative experiments using the following algorithms.

- AMIPGMls-I with non-inertial accelerated ($m = 0$ in Algorithm 2);
- AMIPGMls-II with 1-step inertial accelerated ($m = 1$ in Algorithm 2);
- Ordinary PG with stepsize $\frac{1}{L}$;
- IPGM in [11], whose inertial factor $\omega = 0.8$, and stepsize $\gamma^k = \frac{0.999*(2-2\omega)}{L}$;
- iPiano in [16], whose inertial factor $\omega = 0.8$, and stepsize $\gamma^k = \frac{0.999*(2-2\omega)}{L}$.

For the AMIGls-I and II, we use the parameter as follows, $q = 0.6$, $\eta = 0.8$, $\omega_{\max} = 0.95$, $\theta = 0.22$, $\sigma = 10^{-4}$, $\tau = 2$, $c = 1$, $\gamma_{\max} = \frac{10}{\max\{L, 10^{-12}\}}$, $L_{\min} = 10^{-12}$, $\varepsilon = 10^{-12}$.

In this experiment, the initial point is chosen as $x^0 = 0$. we run all algorithms for following terminate condition

$$\frac{\|x^{k+1} - x^k\|}{\max\{\|x^{k+1}\|, 1\}} \leq 10^{-6} \quad \text{or} \quad k > 1000.$$

In Table 2, We report results with following performance metrics: iteration numbers (Iter), final objective value (Obj), $\log(\|G_{\gamma^k}(x^k)\|)$ and total running time in seconds (Time (s)).

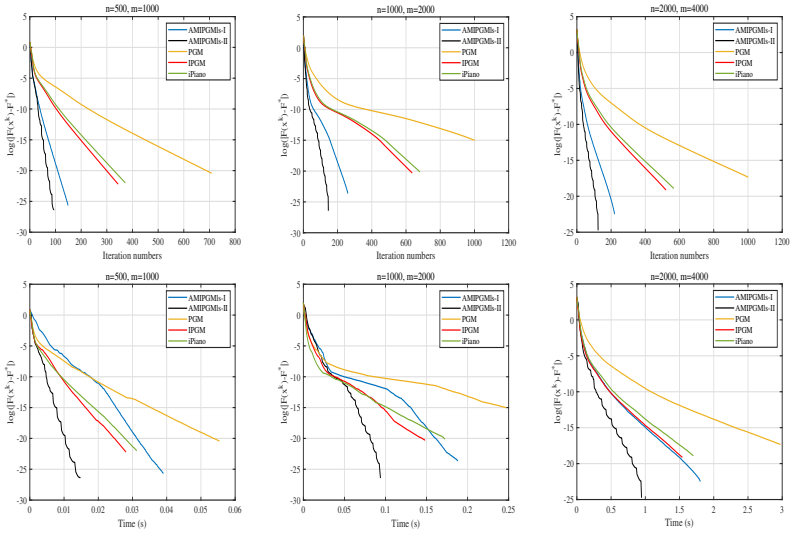


Fig. 2: Comparison of different algorithms

$(n = 500, m = 1000)$	Iter	Obj	$\log(\|G_{\gamma^k}(x^k)\|)$	Time (s)
AMIPGMls-I	125	2.2379	-12.6992	0.015
AMIPGMls-II	92	2.2379	-13.7053	0.011
PGM	709	2.2379	-11.2947	0.046
IPGM	344	2.2379	-12.1772	0.032
iPiano	372	2.2379	-12.0755	0.034
$(n = 1000, m = 2000)$				
AMIPGMls-I	220	4.5142	-14.0360	0.085
AMIPGMls-II	154	4.5142	-12.2741	0.062
PGM	1000	4.5142	-8.8664	0.218
IPGM	636	4.5142	-11.4912	0.139
iPiano	681	4.5142	-11.3974	0.174
$(n = 2000, m = 4000)$				
AMIPGMls-I	191	9.1093	-11.6054	0.988
AMIPGMls-II	147	9.1093	-12.3912	0.742
PGM	1000	9.1093	-10.0611	2.896
IPGM	521	9.1093	-10.9193	1.453
iPiano	566	9.1093	-10.8172	1.665

Table 2: Experimental results with different (m, n)

To compare the convergence behavior more clearly, we also plot the objective gap to the best final value among all methods (denote $\log(|f(x^k) - f^*|)$) along with iteration numbers in Figure 2. The results show that the AMIPGMls also exhibits favorable numerical performance in the absence of inertia. Compared with IPGM and iPiano, which incorporate inertial acceleration but employ fixed-parameter updates, AMIPGMls adopts a more aggressive adaptive local-curvature strategy and with the aid of line search, achieves safe and rapid convergence. After introducing inertia, the numerical performance of AMIPGMls is further improved, which also confirms the effectiveness of inertial acceleration.

6 Conclusion

We proposed adaptive multi-step inertial gradient and proximal gradient methods for nonconvex optimization. The methods use local curvature information, multi-step extrapolation, and a sufficient-descent line search. The inertial term is scaled by the current stationarity measure, which reduces the extrapolation near critical points and allows larger steps away from stationarity. For the composite case, the same mechanism is combined with the proximal mapping.

We proved finite termination of the line search, monotone decrease of the objective values, and convergence of the gradient norm or proximal gradient mapping to zero. We also derived an $O(\epsilon^{-2})$ iteration complexity bound. The numerical experiments are consistent with the theory and show that the proposed methods compare favorably with the tested first-order baselines.

Declarations

Funding This work is supported by the National Natural Science Foundation of China (No. 72471227) and the Graduate Innovation Program of China University of Mining and Technology (Project No. KYCX25_2856).

Author contributions All authors read and approved the final manuscript. X.Q. Wang was primarily responsible for the algorithm design, theoretical analysis, numerical experiments, and manuscript preparation. H. Shao contributed to the algorithm design and manuscript revision.

Conflicts of interest The authors declared that they have no conflicts of interest to this work.

Data availability No datasets were generated or analyzed during the current study.

References

[1] Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka–Lojasiewicz inequality. *Math. Oper. Res.* **35**(2), 438–457 (2010).

[2] Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2**(1), 183–202 (2009).

[3] Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.* **146**(1–2), 459–494 (2014).

[4] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2011).

[5] Cai, J.-F., Candès, E. J., Shen, Z.: A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **20**(4), 1956–1982 (2010).

[6] Candès, E. J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? *J. ACM* **58**(3), 1–37 (2011).

[7] Chartrand, R., Staneva, V.: Restricted isometry properties and nonconvex compressive sensing. *Inverse Probl.* **24**(3), 035020 (2008).

[8] Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96**(456), 1348–1360 (2001).

[9] Gotoh, J., Takeda, A., Tono, K.: DC formulations and algorithms for sparse optimization problems. *Math. Program.* **169**(1), 141–176 (2018).

[10] Kim, S.-J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE J. Sel. Top. Signal Process.* **1**(4), 606–617 (2007).

[11] László, S. C.: Convergence rates for an inertial algorithm of gradient type associated to a smooth nonconvex minimization. *Math. Program.* **190**, 285–329 (2021).

[12] Lou, Y., Yin, P., He, Q., Xin, J.: Computing sparse representation in a highly coherent dictionary based on difference of ℓ_1 and ℓ_2. *J. Sci. Comput.* **64**, 178–196 (2015).

[13] Lou, Y., Yan, M.: Fast ℓ_1–ℓ_2 minimization via a proximal operator. *J. Sci. Comput.* **74**(2), 767–785 (2018).

[14] Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR* **269**(3), 543–547 (1983).

[15] Nesterov, Y.: *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, New York (2004).

[16] Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: inertial proximal algorithm for nonconvex optimization. *SIAM J. Imaging Sci.* **7**, 1388–1419 (2014).

[17] Polyak, B. T.: Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.* **4**, 1–17 (1964).

[18] Polyak, B. T.: *Introduction to Optimization*. Optimization Software Inc., New York (1987).

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

[20] Rockafellar, R. T., Wets, R. J.-B.: *Variational Analysis*. Springer, Berlin (1998).

[21] Wen, B., Chen, X., Pong, T. K.: A proximal difference-of-convex algorithm with extrapolation. *Comput. Optim. Appl.* **69**(2), 297–324 (2018).

[22] Wu, Z. M., Li, M.: General inertial proximal gradient method for a class of nonconvex nonsmooth optimization problems. *Comput. Optim. Appl.* **73**, 129–158 (2019).

[23] Wu, Z. M., Li, C. S., Li, M.: Inertial proximal gradient methods with Bregman regularization for a class of nonconvex optimization problems. *J. Glob. Optim.* **79**, 617–644 (2021).

[24] Zeng, L. M., Xie, J.: Group variable selection via SCAD-ℓ_2. *Statistics* **48**, 49–66 (2014).