Bregman Proximal Gradient Algorithm with Extrapolation for a class of Nonconvex Nonsmooth Minimization Problems

Xiaoya Zhang*, Roberto Barrio† §, M. Angeles Martínez‡§, Hao Jiang¶, Lizhi Cheng* ¶

Preprint submitted for publication, February 14, 2019

Abstract

In this paper, we consider an accelerated method for solving nonconvex and nonsmooth minimization problems. We propose a Bregman Proximal Gradient algorithm with extrapolation (BPGe). This algorithm extends and accelerates the Bregman Proximal Gradient algorithm (BPG), which circumvents the restrictive global Lipschitz gradient continuity assumption needed in Proximal Gradient algorithms (PG). The BPGe algorithm has higher generality than the recently introduced Proximal Gradient algorithm with extrapolation (PGe), and besides, due to the extrapolation step, BPGe converges faster than BPG algorithm. Analyzing the convergence, we prove that any limit point of the sequence generated by BPGe is a stationary point of the problem by choosing parameters properly. Besides, assuming Kurdyka-Lojasiewicz property, we prove the whole sequences generated by BPGe converges to a stationary point. Finally, to illustrate the potential of the new method BPGe, we apply it to two important practical problems that arise in many fundamental applications (and that not satisfy global Lipschitz gradient continuity assumption): Poisson linear inverse problems and quadratic inverse problems. In the tests the accelerated BPGe algorithm shows faster convergence results, giving an interesting new algorithm.

Keywords: Bregman Proximal Gradient algorithm with extrapolation, Bregman Distance, Proximal Gradient Algorithm, Smooth Adaptive Condition, Relative Weakly Convexity

MSC codes: 90C30, 90C26, 47N10

1 Introduction

In the last few years different numerical methods have been devised to solve large-scale minimization problems, but still the Cauchy’s gradient method is at the kernel of most of the schemes (for

*Department of Mathematics, National University of Defense Technology, Changsha, 410073, Hunan, China. (Email: zhangxiaoya09, clzcheng@nudt.edu.cn)
†Departamento de Matemática Aplicada and IUMA. Computational Dynamics group. University of Zaragoza. E-50009, Spain. (Email: rbarrio@unizar.es)
‡College of Computer, National University of Defense Technology, Changsha, 410073, Hunan, China. (Email: haojiang@nudt.edu.cn)
§The authors X.Z., H.J. and L.C. were supported by the National Key Research and Development Program of China (NO.2017YFB0202003), National Natural Science Foundation of Hunan (2018JJ3616).
¶The authors R.B. and A.M. were supported by the Spanish Research projects MTM2015-64095-P, PGC2018-096026-B-I00, and European Regional Development Fund and Diputación General de Aragón (E24-17R).
instance, see the recent books [1, 2] and it is assumed that the gradient of the objective function is globally Lipschitz continuous. This assumption is quite restrictive in some real applications, and therefore recently new families of methods have been designed in order to solve more generic cases. On this line, the remarkable paper of Bauschke, Bolte and Teboulle [3] introduced a new method based on the Bregman distance paradigm (BPG algorithm) able to deal with non-globally Lipschitz continuous gradient problems in the convex case, and Bolte, Sabach, Reboulle and Vaisbourd [4] extend it to the nonconvex case.

On the other hand, a lot of effort has been paid to accelerate the proximal gradient algorithm in order to reduce the number of iterations. Several techniques have been introduced, like the fast iterative shrinkage-thresholding algorithm (FISTA) proposed in [5], the use of Nesterov’s extrapolation techniques [6, 7], and quite recently it has been introduced in [8] a version of the proximal gradient algorithm with extrapolation for some nonconvex nonsmooth minimization problems (but assuming that the gradient of the objective function is globally Lipschitz continuous).

The main goal of this paper is to focus on introducing a scheme, and analyzing the convergence, that combines the power of the method developed in [4] able to solve non-globally Lipschitz continuous gradient problems in the convex and nonconvex case, and that includes extrapolation techniques [8] in order to accelerate the method.

In this paper, we consider the following minimization problem:

$$\inf \{ \Psi(x) := f(x) + g(x) : x \in \mathbb{R}^d \}.$$  \hfill (P)

where $f$ is a nonconvex continuously differentiable function and $g$ is a proper lower-semi-continuous (l.s.c.) convex function. We assume that the optimal value of (P) is finite, that is, $\Psi^* := \inf \{ \Psi(u) : u \in \mathbb{R}^d \} > -\infty$. Problem (P) arises in many applications including compressed sensing [9], signal recovery [10], phase retrieve problem [11]. One classical algorithm for solving this problem is the Proximal Gradient (PG) method [12]:

$$x^{k+1} = \arg \min_x \left\{ g(x) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2\lambda_k} \|x - x^k\|^2 \right\}, k \in \mathbb{N},$$

where $\lambda_k$ is the stepsize on each iteration. Proximal gradient method and its variants [13, 14, 15, 16, 17, 18] have been one hot topic in optimization field for a long time due to their simple forms and lower computation complexity.

One branch of developing new PG methods was devoted to accelerations. Accelerated proximal algorithms [5, 19] on convex problems have shown to be quite efficient. They were also useful for solving nonconvex problems [3, 20, 21, 22]. For solving nonconvex problems (P), one simple and efficient strategy is to perform extrapolation for each $k \in \mathbb{N}$, with the following form (where $x^{-1} = x^0$)

$$\begin{cases} y^k = x^k + \beta_k(x^k - x^{k-1}), \\ x^{k+1} = \arg \min_x \left\{ g(x) + \langle \nabla f(y^k), x - y^k \rangle + \frac{1}{2\lambda_k} \|x - y^k\|^2 \right\}, \end{cases}$$

where $\lambda_k$ is the stepsize on each iteration, and $\beta_k(x^k - x^{k-1})$ is an extrapolation term. The previous iteration is called the Proximal Gradient algorithm with Extrapolation (PGe), which have been shown in [8] that converges and performs quite well by setting parameters $\beta_k$ properly. However, PGe has one restriction on solving problem (P): it requires the continuously differentiable part $f$ to
be globally Lipschitz gradient continuous on \( \mathbb{R}^d \). In fact, this requirement cannot often be satisfied for many practical problems, such as quadratic inverse problem in phase retrieve [11] and Poisson linear inverse problems [23], that arise in many real world applications.

In this paper, we propose a new algorithm — Bregman Proximal Gradient algorithm with Extrapolation (BPGe) — to solve problem (P) without requiring globally Lipschitz gradient continuity of \( f \) for each \( k \in \mathbb{N} \), from \( x^{-1} = x^0 \):

\[
\begin{align*}
    y^k &= x^k + \beta_k (x^k - x^{k-1}), \\
    x^{k+1} &= \arg \min_x \left\{ g(x) + \langle \nabla f(y^k), x - y^k \rangle + \frac{1}{\lambda_k} D_h(x, y^k) \right\},
\end{align*}
\]

where \( D_h \) is a Bregman distance defined in Section 2. On the basis of Bregman distance theory, we utilize a smooth adaptive condition introduced in [4], which generalizes Lipschitz gradient continuous condition. This smooth adaptive condition was originally proposed to analyze Bregman Proximal Gradient (BPG) algorithm in [4]. It can also be used to analyze the convergence of BPGe, since BPGe algorithm extends BPG one by performing extrapolation. In particular, we have that:

(i) When \( D_h(x, y) = \frac{1}{2} \| x - y \|^2 \) and \( \beta_k = 0 \), BPGe reduces to PG.

(ii) When \( D_h(x, y) = \frac{1}{2} \| x - y \|^2 \), BPGe reduces to PGe;

(iii) When \( \beta_k = 0 \) for any \( k \geq 0 \), BPGe reduces to BPG (no extrapolation).

Therefore, PG, PGe and BPG are particular cases of BPGe algorithm.

Recently, other acceleration algorithms for BPG have been proposed in literature, like using it combined with inertial methods [24] (which is a different methodology from ours), or combining it with Nesterov’s acceleration method [25] but requiring the Bregman distance function satisfying some extra crucial triangle scaling property.

From the convergence analysis (Section 4), the BPGe algorithm has to satisfy the condition \( D_h(x^k, y^k) \leq \rho C_k D_h(x^{k-1}, x^k) \) (where \( C_k \in (0, 1) \) and \( \rho \in (0, 1) \) are two parameters) to guarantee the convergence. In the Lipschitz gradient continuous condition \( D_h(x, y) = \frac{1}{2} \| x - y \|^2 \), this condition is easily satisfied just by choosing \( \inf_{k \in \mathbb{N}} \{ \beta_k \} \leq \sqrt{\rho C} \). But when \( D_h \) is general, computing a threshold of \( \inf_{k \in \mathbb{N}} \{ \beta_k \} \) directly may be hard and expensive. Therefore, we modify this idea to achieve this condition through a line search method (Algorithm 2 introduced in Section 3).

In the convergence analysis, we prove that any limit point of the sequence generated by BPGe is a stationary point under very general conditions. Moreover, by adding some slightly stronger assumptions and Kurdyka-Lojasiewicz property, we could guarantee the sequence generated by BPGe converges to a stationary point.

The paper is organized as follows. We first introduce in Section 2 some basic definitions in optimization, smooth adaptive condition, relative weak convexity, and Kurdyka-Lojasiewicz property. In Section 3 we introduce the new BPGe algorithm. The convergence analysis is done in Section 4, where under some assumptions of the smooth adaptive condition and relative weak convexity of problem (P), we first show a descent-type lemma, from which the fact that any limit point of the sequence generated by BPGe is a critical point follows. Later, we prove that the whole sequence generated by BPGe converges to a critical point under Kurdyka-Lojasiewicz property and a stronger assumption. Several numerical experiments are shown in Section 5 to show the performance of the BPGe method compared with the BPG one.
2 Preliminaries

Throughout the paper we will use the following basic notations. Let \( \mathbb{N} := \{0, 1, 2, \ldots \} \) be the set of nonnegative integers. We will always work in the Euclidean space \( \mathbb{R}^d \), and the standard Euclidean inner product and the induced norm on \( \mathbb{R}^d \) are denoted by \( \langle \cdot, \cdot \rangle \) and \( \| \cdot \| \), respectively. We denote \( B_\rho(\tilde{x}) := \{x \in \mathbb{R}^d : \|x - \tilde{x}\| \leq \rho\} \) as the ball of radius \( \rho > 0 \) around \( \tilde{x} \in \mathbb{R}^d \), \( \text{dist}(x,S) := \inf_{y \in S} \|x - y\| \) as the distance from a point \( x \in \mathbb{R}^d \) to a nonempty set \( S \subset \mathbb{R}^d \). The domain of the function \( f : \mathbb{R}^d \rightarrow (-\infty, +\infty] \) is defined by \( \text{dom} f := \{x \in \mathbb{R}^d : f(x) < +\infty\} \). We say that \( f \) is proper if \( \text{dom} f \neq \emptyset \). For other generalized notions and definitions we refer to [4, 26, 27].

2.1 Smooth Adaptable Function and Relative Weakly Convexity

In this subsection, we define the notion of smooth adaptable condition for nonconvex \( f \) proposed in [4]. This property was extended from the recent work [3] in which the differentiable functions need to be convex. This condition is similar to the relative smoothness condition introduced in [28], but the relative smoothness is based on the fact that \( f \) is convex. As we want also to deal with nonconvex functions, in our paper we use the smooth adaptable condition to generalize Lipschitz gradient continuity and to derive the related convergence results of BPGe.

We first introduce the concept of Bregman distance needed in the definition of smooth adaptable condition.

**Definition 1.** (Kernel Generating Distance and Bregman Distance [4]) Let \( S \) be a nonempty, convex and open subset of \( \mathbb{R}^d \). Associated with \( S \), a function \( h : \mathbb{R}^d \rightarrow (-\infty, +\infty] \) is called a kernel generating distance if it satisfies the following:

(i) \( h \) is proper, lower-semi-continuous and convex, with \( \text{dom} h \subset S \) and \( \text{dom} \partial h = S \).

(ii) \( h \) is \( C^1 \) on \( \text{int} \text{ dom} h \equiv S \).

We denote the class of kernel generating distances by \( \mathcal{G}(S) \). Given \( h \in \mathcal{G}(S) \), the Bregman distance [29] is defined by \( D_h : \text{dom} h \times \text{int} \text{ dom} h \rightarrow [0, +\infty) \)

\[
D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle.
\]

Note that the Bregman distance is, obviously, a proximity measure that measures the proximity of \( x \) and \( y \). Next, we list some basic properties of the Bregman distance [30, 31]:

(i) For any \( (x, y) \in \text{dom} h \times \text{int} \text{ dom} h \), \( D_h(x, y) \geq 0 \). If in addition \( h \) is strictly convex, \( D_h(x, y) = 0 \) if and only if \( x = y \) holds.

(ii) The three point identity: For any \( y, z \in \text{int} \text{ dom} h \) and \( x \in \text{dom} h \),

\[
D_h(x, z) - D_h(x, y) - D_h(y, z) = \langle \nabla h(y) - \nabla h(z), x - y \rangle.
\]

(iii) Linear Additivity: For any \( \alpha, \beta \in \mathbb{R} \), and any functions \( h_1 \) and \( h_2 \) we have:

\[
D_{\alpha h_1 + \beta h_2}(x, y) = \alpha D_{h_1}(x, y) + \beta D_{h_2}(x, y),
\]

for all couple \( (x, y) \in (\text{dom} h_1 \cap \text{dom} h_2)^2 \) such that both \( h_1 \) and \( h_2 \) are differentiable at \( y \).
Throughout the paper we will focus on the pair of functions \((f, h)\) that satisfies the smooth adaptable condition. Next we present the definition introduced in [1].

**Definition 2.** \((L\text{-smooth adaptable}[4])\) A pair of functions \((f, h)\), such that \(h \in \mathcal{G}(S)\), \(f : \mathbb{R}^d \rightarrow (-\infty, +\infty]\) is a proper and lower-semi-continuous function with \(\text{dom } h \subset \text{dom } f\), which is continuously differentiable on \(S = \text{int } \text{dom } h\), is called \(L\)-smooth adaptable \((L\text{-smad})\) on \(S\) if there exists \(L > 0\) such that \(Lh - g\) and \(Lh + g\) are convex on \(S\).

According to [4, Lemma 2.1], the pair of functions \((f, h)\) is \(L\text{-smad}\) on \(S\) if and only if
\[
\|f(x) - f(y) - \langle \nabla f(y), x - y \rangle\| \leq L D_h(x, y) \quad \text{for any } (x, y) \in \text{int dom } h.
\]

When \(h(x) = \frac{1}{2}\|x\|^2\) and consequently \(D_h(x, y) = \frac{1}{2}\|x - y\|^2\), the \(L\)-smad condition of \(f\) would be reduced to Lipschitz gradient continuity:
\[
\|f(x) - f(y) - \langle \nabla f(y), x - y \rangle\| \leq L^2 \|x - y\|^2 \quad \text{for any } (x, y) \in \text{dom } h.
\]

Next we introduce the definition of a \(\mu\)-relative weakly convex function, given in [32]. This definition extends the definition of weakly convexity [33], which was employed in the analysis of nonconvex optimization methods.

**Definition 3.** \(f\) is called \(\mu\)-relative weakly convex to \(h\) on \(S\) if there exists \(\mu > 0\) such that \(f + \mu h\) is convex on \(S\).

When \(f\) is convex, \(\mu = 0\). When \((f, h)\) is \(L\text{-smad}\) on \(S\), obviously \(f\) is \(L\)-relative weakly convex to \(h\). So by default, \(\mu \leq L\).

### 2.2 Kurdyka–Lojasiewicz Property

Finally, we introduce the definition of the Kurdyka–Lojasiewicz property proposed in [34]. We need this property to prove the global convergence of the whole sequences generated by BPGe for solving (P).

**Definition 4.** \((Kurdyka–Lojasiewicz property)[34]\) Let \(f : \mathbb{R}^d \rightarrow (-\infty, +\infty]\) be a proper lower-semi-continuous function.

(i) The function \(f\) is said to have the Kurdyka–Lojasiewicz (KL) property at \(\bar{x} \in \text{dom } \partial f := \{x \in \mathbb{R}^d : \partial f(x) \neq \emptyset\}\) if there exist \(\eta \in (0, +\infty]\), a neighborhood \(U\) of \(\bar{x}\) and a function \(\psi : (0, \eta) \rightarrow \mathbb{R}_+\) satisfying:
\[
\psi(0) = 0, \psi \in C^1(0, \eta) \text{ and continuous at } 0, \text{ for all } s \in (0, \eta) : \psi'(s) > 0
\]

such that for all \(x \in U \cap [f(\bar{x}) < f(x) < f(\bar{x}) + \eta]\), the following inequality holds
\[
\psi'(f(x) - f(\bar{x})) \cdot \text{dist}(0, \partial f(x)) \geq 1.
\]

(ii) If \(f\) satisfies the KL property at each point of \(\text{dom } \partial f\) then \(f\) is called a KL function.

### 3 Bregman Proximal Gradient Algorithm with Extrapolation (BPGe)

Throughout this paper, we focus on the nonconvex problem (P) in Section 1 with the following assumptions on \(f\) and on the kernel generating distance function \(h\): \(h \in \mathcal{G}(\mathbb{R}^d)\), \((f, h)\) is \(L\text{-smad}\).
and \(f\) is \(\mu\)-weakly convex relative to \(h\) (see Definition 2 and 3). And we also make the following general Assumptions 1 and 2 as default.

**Assumption 1.** The function \(\Psi\) is supercoercive, that is,
\[
\lim_{\|u\| \to \infty} \frac{\Psi(u)}{\|u\|} = \infty.
\]

Assumptions 2 is a general assumption used in the analysis of Bregman Proximal-type algorithms [3, 30].

**Assumption 2.**

(i) \(h\) is strictly convex.

(ii) If \(\{x^k\}_{k \in \mathbb{N}}\) converges to some \(x\) in \(\text{dom } h\) then \(D_h(x, x^k) \to 0\).

(iii) If \(\{x^k\}_{k \in \mathbb{N}}, \{y^k\}_{k \in \mathbb{N}}\) defined in \(\text{dom } h\) are sequences such that \(y^k \to x^* \in \overline{\text{dom } h}\), \(\{x^k\}_{k \in \mathbb{N}}\) is bounded, and if \(D_h(x^k, y^k) \to 0\), then \(x^k \to x^*\).

We are now ready to introduce our BPGe algorithm, divided in two parts, Algorithm 1 and Algorithm 2. Algorithm 1 is the whole framework for solving Problem (P). And Algorithm 2 is a line search step, which is used to search a proper parameter \(\beta_k\) at every iteration in Algorithm 1.

Throughout the whole paper, we make the following notations
\[
\bar{\lambda} := \sup_{k \in \mathbb{N}} \{\lambda_k\}, \quad \underline{\lambda} := \inf_{k \in \mathbb{N}} \{\lambda_k\}.
\]

By default \(0 < \underline{\lambda} \leq \bar{\lambda} < \infty\).

---

**Algorithm 1: BPGe—Bregman Proximal Gradient algorithm with Extrapolation.**

**Data:** A function \(h\) defined in Definition 1 such that \((f, h)\) is \(L\)-smooth and \(f\) is \(\mu\)-weakly convex relative to \(h\) on \(\mathbb{R}^d\).

Error tolerance: TOL.

**Initialization:** \(x^0 = x^{-1} \in \text{int } \text{dom } h\) and \(0 < \lambda_k \leq 1/L\).

**General step:**

For \(k = 0, 1, 2, \ldots, k_{\max}\) repeat

Take
\[
y^k = x^k + \beta_k (x^k - x^{k-1}),
\]

where \(\beta_k\) is searched according to **Line Search** in Algorithm 2.

Then compute
\[
x^{k+1} \in \arg\min \left\{ x : g(x) + \left\langle x - y^k, \nabla f(y^k) \right\rangle + \frac{1}{\lambda_k} D_h(x, y^k), x \in \mathbb{R}^d \right\}.
\]

until EXIT(TOL) received.
Algorithm 2: Line Search for Algorithm 1 at the $k$-th iteration.

**Data:** A function $h$ defined in Algorithm 1, fix $0 < \eta < 1$, $\beta_0 \in [0, 1)$, $0 < \rho < 1$.

**Input:** $x^{k-1}, x^k \in \text{int dom } h$, $C_k = \frac{\lambda_k^{-1}}{\lambda_k^{k+1}}$.

**General step:**

$(\tilde{\beta} = \beta_0$)

While $D_h(x^k, x^k + \tilde{\beta}(x^k - x^{k-1})) > \rho C_k D_h(x^{k-1}, x^k)$ do

$\tilde{\beta} = \eta \tilde{\beta}$.

**Return:** Set the feasible step size $\beta_k = \tilde{\beta}$ for iteration $k$.

We remark that an important point on any iterative process is to define suitable error control conditions. On one hand we fix a maximum number of iterations $k_{\text{max}}$ (in most of our tests 5000 iterations) and EXIT(TOL)=true if $\|x^k - x^{k-1}\|/\max\{1, \|x^k\|\} \leq \text{TOL}$ (in our tests TOL = 10$^{-6}$ as in [8]). Other option is to check the convergence using the objective function, instead of the solution itself, that is $\|\Psi(x^k) - \Psi(x^{k-1})\|/\max\{1, \|\Psi(x^k)\|\} \leq \text{TOL}$.

We first verify that (2) is well-defined using the following Proposition 1. For all $y \in \text{int dom } h$ and stepsize $0 < \lambda \leq 1/L$, we define the Bregman proximal gradient mapping as:

$$T_\lambda(y) := \text{arg min} \{ g(u) + \langle \nabla f(y), u - y \rangle + \lambda^{-1} D_h(u, y) : u \in \mathbb{R}^d \}.$$  (3)

In Proposition 1 we prove that $T_\lambda(y)$ is well posed. Thus by Proposition 1, $x^{k+1} \in T_\lambda(x^k)$, and fixing $\inf \{\lambda_k\} > 0$, then Step (2) in BPG algorithm is well-defined.

**Proposition 1.** Suppose that Assumption [4] holds, let $y \in \text{int dom } h$ and $0 < \lambda \leq 1/L$. Then, the set $T_\lambda(y)$ is a nonempty and compact set.

**Proof.** Fix any $y \in \text{int dom } h$ and $0 < \lambda \leq 1/L$. For any $u \in \mathbb{R}^d$, we define

$$\Psi_h(u) = g(u) + f(y) + \langle u - y, \nabla f(y) \rangle + \lambda^{-1} D_h(u, y),$$

so that $T_\lambda(y) = \text{arg min}_{u \in \mathbb{R}^d} \Psi_h(u)$, It can also be represented as

$$\Psi_h(u) = \Psi(u) - f(u) + f(y) + \langle u - y, \nabla f(y) \rangle + \lambda^{-1} D_h(u, y)$$

$$\geq \Psi(u) + L D_h(u, y) - \left[ f(u) - f(y) - \langle u - y, \nabla f(y) \rangle \right]$$

$$\geq \Psi(u).$$

where the second inequality is obtained by taking into account $\lambda^{-1} \geq L$ and in the last inequality that $(f, h)$ is $L$-smooth adaptable. According to Assumption [4] i.e. $\lim_{\|u\| \to \infty} \Psi(u) = \infty$, there is

$$\lim_{\|u\| \to \infty} \Psi_h(u) \geq \lim_{\|u\| \to \infty} \Psi(u) = \infty.$$

Since $\Psi_h$ is also proper and lower-semi-continuous, invoking the modern form of Weierstrass’ theorem (see, e.g., [26] Theorem 1.9, page 11]), it follows that the value $\inf_{\mathbb{R}^d} \Psi_h$ is finite, and the set $\text{arg min}_{u \in \mathbb{R}^d} \Psi_h(u) \equiv T_\lambda(y)$ is nonempty and compact. \qed
Secondly, we add an extrapolation step to the BPGe algorithm to choose a suitable $\beta_k$ at each iteration step through the line search Algorithm 2. On this step it is hard to guarantee directly the decrease of function value $\Psi(x^k)$. Therefore, we focus on guaranteeing sufficient decrease of the Lyapunov sequences defined in Section 4 in the convergence analysis. However, it still requires an extra condition $D_h(x^k, x^k + \beta_k(x^k - x^{k-1})) \leq \rho C_k D_h(x^{k-1}, x^k)$. When $h = \frac{1}{2} \|x\|^2$, BPGe is reduced to the PGe algorithm [8] and this condition is easily satisfied by setting $0 \leq \beta_k \leq \sqrt{\frac{\rho}{L + \mu}}$. But when $h$ is more general and complex, computing the threshold of $\beta_k$ directly may be hard and expensive. So, we try to reach this condition by a line search method introduced in Algorithm 2. Thus, our next step is to verify that Algorithm 2 is well-defined, as the following proposition 2 shows.

**Proposition 2. (Finite termination of Algorithm 2).** Consider Algorithm 1 and fix $k \in \mathbb{N}$. Let $0 < \eta < 1, 0 < \rho < 1$, $\tilde{\beta} \in [0, 1), C_k = \frac{\lambda_k - 1}{\lambda_k + \mu} > 0$. Then, there exists $J \in \mathbb{N}$ such that $\beta_k := \eta^j \tilde{\beta}$ satisfies

$$D_h(x^k, x^k + \beta_k(x^k - x^{k-1})) \leq \rho C_k D_h(x^{k-1}, x^k)$$

for any $j \geq J$.

**Proof.** This result is proved by contradiction. Suppose that

$$D_h(x^k, x^k + \eta^j \tilde{\beta}(x^k - x^{k-1})) > \rho C_k D_h(x^{k-1}, x^k)$$

holds for any $j \in \mathbb{N}$.

When $x^k = x^{k-1}$, Algorithm 2 terminates with $\beta_k = \tilde{\beta}$ directly.

When $x^k \neq x^{k-1}$, since

$$\|x^k - (x^k + \tilde{\beta}(x^k - x^{k-1}))\| = \eta^j \tilde{\beta} \|x^k - x^{k-1}\| \to 0, \quad j \to \infty,$$

according to Assumption 2(ii), $D_h(x^k, x^k + \eta^j \tilde{\beta}(x^k - x^{k-1})) \to 0$. Thus for any $\varepsilon > 0$, there exist a number $J \in \mathbb{N}$ such that

$$D_h(x^k, x^k + \eta^j \tilde{\beta}(x^k - x^{k-1})) < \varepsilon, \quad \text{for all } j \geq J.$$

Since $x^k \neq x^{k-1}$, and due to the strictly convexity of $h$ in Assumption 2(iii),

$$D_h(x^{k-1}, x^k) > 0.$$

If we set $\varepsilon = \frac{1}{2} \rho C_k D_h(x^{k-1}, x^k)$, then

$$\rho C_k D_h(x^{k-1}, x^k) < D_h\left(x^k, x^k + \eta^j \tilde{\beta}(x^k - x^{k-1})\right) < \frac{1}{2} \rho C_k D_h(x^{k-1}, x^k),$$

for $j \geq J$, which is a contradiction. \qed
4 Convergence Analysis of BPGe

In this section we provide the main convergence results of the BPGe algorithm. First of all, following the analysis of Remark 4.1(ii) in [4], we obtain the following Lemma 1. We find that after adding an extrapolation term, it is hard to justify monotonicity of the objective function $\Psi$ directly. But for a special auxiliary sequence, defined by

$$H_{k,M} := \Psi(x^k) + MD_h(x^{k-1}, x^k), \quad M > 0, \quad \forall k \in \mathbb{N}$$

the monotone property will be presented in our settings.

**Lemma 1.** For any $x \in \text{int dom } h$, and let be a sequence $\{x^k\}_{k \in \mathbb{N}}$ produced by BPGe, then

(i) For any $k \in \mathbb{N}$, we have

$$\Psi(x^{k+1}) - \Psi(x) \leq (\lambda_k^{-1} + \mu) D_h(x, y^k) - \lambda_k^{-1} D_h(x, x^{k+1}) - (\lambda_k^{-1} - L) D_h(x^{k+1}, y^k).$$  \quad (4)

(ii) For any $k \in \mathbb{N}$, we have

$$H_{k+1,M} - H_{k,M} \leq (M - \lambda_k^{-1}) D_h(x^k, x^{k+1}) - (M - \rho \lambda_k^{-1}) D_h(x^{k-1}, x^k).$$  \quad (5)

Moreover, assuming there exists some $M$ such that $\rho \lambda^{-1} \leq M \leq \lambda^{-1}$, then the sequence $\{H_{k,M}\}$ is nonincreasing and convergent for the fixed $M$.

**Proof.** (i) According to the first order condition of (2), we get

$$0 \in \partial g(x^{k+1}) + \nabla f(y^k) + \lambda_k^{-1} (\nabla h(x^{k+1}) - \nabla h(y^k)), \quad \forall k \in \mathbb{N}.$$  

Combining with the convexity of $g$, there is

$$g(x) - g(x^{k+1}) \geq \langle - \nabla f(y^k) - \lambda_k^{-1} (\nabla h(x^{k+1}) - \nabla h(y^k)), x - x^{k+1} \rangle, \quad \forall k \in \mathbb{N}.$$  

Together with the three point identity of Bregman distance

$$\lambda_k^{-1} (\nabla h(x^{k+1}) - \nabla h(y^k), x - x^{k+1}) = \lambda_k^{-1} (D_h(x, y^k) - D_h(x, x^{k+1}) - D_h(x^{k+1}, y^k))$$

we have that

$$g(x) - g(x^{k+1}) + f(x) - f(x^{k+1})$$

$$\geq f(x) - f(x^{k+1}) - \langle \nabla f(y^k), x - x^{k+1} \rangle$$

$$- \lambda_k^{-1} \left( D_h(x, y^k) - D_h(x, x^{k+1}) - D_h(x^{k+1}, y^k) \right), \quad \forall k \in \mathbb{N}.$$  \quad (6)

If we take the $\mu$-relative weakly convex property and $L$-smad property of $(f, h)$ (see Definitions 2 and 3),

$$f(x) - f(x^{k+1}) - \langle \nabla f(y^k), x - x^{k+1} \rangle$$

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

$$\geq -\mu D_h(x, y^k) - L D_h(x^{k+1}, y^k), \quad \forall k \in \mathbb{N}.$$  \quad (7)
Thus

\[ \Psi(x^{k+1}) - \Psi(x) \leq (\lambda_k^{-1} + \mu) D_h(x, y^k) - \lambda_k^{-1} D_h(x, x^{k+1}) - (\lambda_k^{-1} - L) D_h(x^{k+1}, y^k). \]

(ii) For any \( k \in \mathbb{N} \), taking \( x = x^k \) into (4), together with \( L \leq \lambda_k^{-1}, D_h(x^{k+1}, y^k) \geq 0 \) we get

\[ \Psi(x^{k+1}) - \Psi(x) \leq (\lambda_k^{-1} + \mu) D_h(x^k, y^k) - \lambda_k^{-1} D_h(x^k, x^{k+1}). \]

If \( x^k = x^{k-1} \), we get \( y^k = x^k \), thus \( D_h(x^k, y^k) = D_h(x^{k-1}, x^k) = 0 \) and

\[ \Psi(x^{k+1}) + \lambda_k^{-1} D_h(x^k, x^{k+1}) \leq \Psi(x^k) + (\lambda_k^{-1} + \mu) \rho C_k D_h(x^{k-1}, x^k). \]  

(8)

If \( x^k \neq x^{k-1} \), according to Algorithm 2, we have \( D_h(x^k, y^k) \leq \rho C_k D_h(x^{k-1}, x^k) \), thus

\[ \Psi(x^{k+1}) + \lambda_k^{-1} D_h(x^k, x^{k+1}) \leq \Psi(x^k) + (\lambda_k^{-1} + \mu) \rho C_k D_h(x^{k-1}, x^k). \]  

(9)

Combining these two cases, we obtain

\[ \Psi(x^{k+1}) + \lambda_k^{-1} D_h(x^k, x^{k+1}) \leq \Psi(x^k) + (\lambda_k^{-1} + \mu) \rho C_k D_h(x^{k-1}, x^k), \quad \forall k \in \mathbb{N}. \]

From the definition of \( H_{k,M} \), we see that

\[ H_{k+1,M} - H_{k,M} \leq (M - \lambda_k^{-1}) D_h(x^k, x^{k+1}) - (M - \rho \lambda_k^{-1}) D_h(x^{k-1}, x^k), \quad \forall k \in \mathbb{N}. \]

Furthermore, assuming there exists some \( M \) such that

\[ \rho \lambda_k^{-1} \leq \rho \lambda_k^{-1} \leq M \leq \lambda_k^{-1}, \]

and fixing one of such values of \( M \), we find that

\[ H_{k+1,M} - H_{k,M} \leq 0, \quad \forall k \in \mathbb{N}, \]

that is, \( \{H_{k,M}\}_{k \in \mathbb{N}} \) is nonincreasing for the fixed value of \( M \).

Recall that \( H_{k,M} \geq \inf \Psi > -\infty \) and \( H_{k,M} \) is nonincreasing. This implies that \( \{H_{k,M}\} \) is convergent for some fixed \( M \).

The next corollary is an obvious result based on Lemma 1. We analyze the boundness of the sequences produced by BPGe algorithm. Since \( H_{k,M} \) is nonincreasing according to Lemma 1(ii), it is easy to verify that the sequence \( \{x^k\}_{k \in \mathbb{N}} \) generated by BPGe is bounded according to Assumption 1. The boundness would act as a tool in the following analysis, so we present this result as the auxiliary Corollary 1:

**Corollary 1.** Assume there exists some \( M \) such that \( \rho \lambda_k^{-1} \leq M \leq \lambda_k^{-1} \), then the sequence \( \{x^k\}_{k \in \mathbb{N}} \) generated by BPGe is bounded.
If the stepsize $\lambda_k$ and parameter $\rho$ in Algorithm 2 satisfy $\rho < \frac{1}{\lambda_k} = \frac{1}{\lambda}$, then we could get sufficient decrease of the auxiliary sequence $\{H_{k,M}\}_{k \in \mathbb{N}}$ for the fixed $M$ given in Lemma [1]. As a consequence, we can bound the sum of Bregman distance between two iteration points generated by BPGe. Moreover, adding stronger assumptions than Assumption [2] on the kernel generating distance $h$, such as strong convexity, we could get that $\lim_{k \to \infty} \|x^k - x^{k-1}\| = 0$ for the sequence $\{x^k\}_{k \in \mathbb{N}}$ in $\mathbb{R}^d$ by BPGe. In this paper, we just consider the set of weaker blanket Assumptions [1] and [2] that permit us to prove that any limit point of the sequence $\{x^k\}_{k \in \mathbb{N}}$ generated by BPGe, if exists, is a stationary point of the objective function $\Psi$.

Assume that $\{x^k\}_{k \in \mathbb{N}}$ is generated from a starting point $x^0$. The set of all limit points of $\{x^k\}_{k \in \mathbb{N}}$ is denoted by

$$\omega(x^0) := \{x : \text{an increasing sequence of integers } \{k_i\}_{i \in \mathbb{N}} \text{ such that } x^{k_i} \to x \text{ as } i \to \infty\}.$$ 

The next technical lemma shows, among other results, that for any $\{x^k\}_{k \in \mathbb{N}}$, $\omega(x^0) \subseteq \text{crit } \Psi$ holds.

**Lemma 2.** Suppose $\rho < \frac{1}{\lambda}$ and let $\{x^k\}_{k \in \mathbb{N}}$ be a sequence generated from $x^0$ by BPGe. Then the following statements hold:

(i) $\sum_{k=0}^{\infty} D_h(x^{k-1}, x^k) < \infty$ and $\lim_{k \to \infty} D_h(x^{k-1}, x^k) = 0$.

(ii) Any limit point of $\{x^k\}_{k \in \mathbb{N}}$ is a critical point of $\Psi (\omega(x^0) \subseteq \text{crit } \Psi)$.

(iii) $\zeta := \lim_{k \to \infty} \Psi(x^k)$ exists and $\Psi \equiv \zeta$ on $\omega(x^0)$.

**Proof.** (i) Since $\rho < \frac{1}{\lambda}$, we have that $\rho \lambda_k^{-1} \leq \frac{1}{\lambda} \lambda_k^{-1} \leq \lambda_k^{-1}$, and we choose $M \in (\rho \lambda_k^{-1}, \lambda_k^{-1}]$. From [5], together with the nonnegativeness of $D_h(x^k, x^{k+1})$ and $M \leq \lambda_k^{-1}$, we have $\forall k \in \mathbb{N}$

$$\left(M - \rho \lambda_k^{-1}\right) D_h(x^{k-1}, x^k) \leq \left(M - \rho \lambda_k^{-1}\right) D_h(x^{k-1}, x^k) \leq H_{k,M} - H_{k+1,M},$$

which implies, $\forall K \in \mathbb{N}$, that

$$0 \leq \sum_{i=0}^{K} \left(M - \rho \lambda_k^{-1}\right) D_h(x^{k-1}, x^k) \leq H_{0,M} - H_{K+1,M},$$

by summing both sides of (10) from 0 to $K$. Since $\{H_{k,M}\}$ is convergent by Lemma [1](ii), letting $K \to \infty$, we conclude that the infinite sum exists and is finite, i.e.,

$$\sum_{i=0}^{K} \left(M - \rho \lambda_k^{-1}\right) D_h(x^{k-1}, x^k) < \infty.$$ 

Since $M - \rho \lambda_k^{-1} > 0$, we obtain directly that $\sum_{i=0}^{K} D_h(x^{k-1}, x^k) \leq \infty$ and $\lim_{k \to \infty} D_h(x^{k-1}, x^k) = 0$.

(ii) Let $\overline{x}$ be a limit point of $\{x^k\}_{k \in \mathbb{N}}$. Let $\{x^{k_i}\}$ be a subsequence such that $\lim_{i \to \infty} x^{k_i} = \overline{x}$. Since $D_h(x^{k_i-1}, x^{k_i}) \to 0$, and we know $\{x^{k_i-1}\}_{i \in \mathbb{N}}$ is bounded according to Corollary [1], Assumption [1](ii) implies $x^{k_i-1} \to \overline{x}$. Similarly, we get $x^{k_i-2} \to \overline{x}$. By the representation of $y^{k_i-1} = x^{k_i-1} + \beta_{k_i-1}(x^{k_i-1} - x^{k_i-2})$ or $y^{k_i-1} = x^{k_i-1}$ (if $x^{k_i-1} = x^{k_i-2}$), we obtain

$$\|y^{k_i-1} - x^{k_i}\| \leq \|x^{k_i-1} - x^{k_i}\| + \|x^{k_i-1} - x^{k_i-2}\| \leq \|x^{k_i} - \overline{x}\| + 2\|x^{k_i-1} - \overline{x}\| + \|x^{k_i-2} - \overline{x}\| \to 0.$$ (12)
On one hand, we prove that there exists \( v^{k_i} \in \partial \Psi(x^{k_i}) \) such that \( v^{k_i} \to 0 \). By using the first-order optimality condition of the minimization problem \([2]\), we obtain

\[
0 \in \lambda_{k_i-1} g(x^{k_i}) + \lambda_{k_i-1} \nabla f(y^{k_i-1}) + \nabla h(x^{k_i}) - \nabla h(y^{k_i-1}), \quad \forall k_i \in \mathbb{N}.
\]

Therefore, we observe that

\[
\nabla f(x^{k_i}) - \nabla f(y^{k_i-1}) - \lambda_{k_i-1}^{-1} (\nabla h(x^{k_i}) - \nabla h(y^{k_i-1})) \in \partial \Psi(x^{k_i}), \quad \forall k_i \in \mathbb{N}. \tag{13}
\]

Taking limits on the left hand in (13) we have that

\[
\| \nabla f(x^{k_i}) - \nabla f(y^{k_i-1}) - \lambda_{k_i-1}^{-1} (\nabla h(x^{k_i}) - \nabla h(y^{k_i-1})) \| \\
\leq \| \nabla f(x^{k_i}) - \nabla f(y^{k_i-1}) \| + \lambda_{k_i-1}^{-1} \| \nabla h(x^{k_i}) - \nabla h(y^{k_i-1}) \| \to 0, \quad k_i \to \infty, \tag{14}
\]

where the limit can be got according to (12) and the continuity of \( \nabla f \) and \( \nabla h \). Thus, we get that there exist \( v^{k_i} \in \partial \Psi(x^{k_i}) \) such that \( \| v^{k_i} \| \to 0 \) as \( k_i \to \infty \).

On the other hand, we derive that \( \Psi(x^{k_i}) \to \Psi(x), k_i \to \infty \). From the lower-semi-continuity of \( \Psi \), we have

\[
\Psi(x) \leq \lim \inf_{i \to \infty} \Psi(x^{k_i}). \tag{15}
\]

According to the iteration step \([2]\) of BPGe, for \( k_i \geq 1 \), we have

\[
\lambda_{k_i-1} g(x^{k_i}) + \langle x^{k_i} - x, \lambda_{k_i-1} \nabla f(y^{k_i-1}) \rangle + D_h(x^{k_i}, y^{k_i-1}) \\
\leq \lambda_{k_i-1} g(x) + D_h(x, y^{k_i-1}).
\]

Adding \( \lambda_{k_i-1} f(x^{k_i}) \) to both sides, we have

\[
\lambda_{k_i-1} \Psi(x^{k_i}) + \langle x^{k_i} - x, \lambda_{k_i-1} \nabla f(y^{k_i-1}) \rangle + D_h(x^{k_i}, y^{k_i-1}) \\
\leq \lambda_{k_i-1} g(x) + \lambda_{k_i-1} f(x^{k_i}) + D_h(x, y^{k_i-1}), \quad \forall k_i \in \mathbb{N}. \tag{16}
\]

After rearranging terms, for all \( k_i \in \mathbb{N} \), it follows

\[
\Psi(x^{k_i}) \leq \Psi(x) + f(x^{k_i}) - f(x) - \langle x^{k_i} - x, \nabla f(y^{k_i-1}) \rangle - \lambda_{k_i-1}^{-1} D_h(x^{k_i}, y^{k_i-1}) + \lambda_{k_i-1}^{-1} D_h(x, y^{k_i-1}). \tag{17}
\]

L-smad property and \( \mu \)-relative weakly convexity of \( (f, h) \) imply that for all \( k_i \in \mathbb{N} \)

\[
f(x^{k_i}) - f(x) - \langle x^{k_i} - x, \nabla f(y^{k_i-1}) \rangle \\
\leq L D_h(x^{k_i}, x) + \langle x^{k_i} - x, \nabla f(x) - \nabla f(y^{k_i-1}) \rangle \\
= L D_h(x^{k_i}, x) + D_f(x^{k_i}, y^{k_i-1}) - D_f(x, y^{k_i-1}) - D_f(x^{k_i}, x) - D_f(x, y^{k_i-1}) \\
\leq L D_h(x^{k_i}, x) + L D_h(x^{k_i}, y^{k_i-1}) + \mu D_h(x^{k_i}, x) + \mu D_h(x, y^{k_i-1}) \tag{18}
\]
Plugging (18) in (17), passing to the limit, together with the relationship $\lambda \leq \lambda_i \leq \bar{\lambda}$, we have
\[
\lim_{i \to \infty} \Psi(x^{k_i}) \leq \Psi(\bar{x}) + \lim_{i \to \infty} \left[ (-\bar{\lambda}^{-1} + L)D_h(x^{k_i}, y^{k_i-1}) + (\lambda^{-1} + \mu)D_h(\bar{x}, y^{k_i-1}) + (L + \mu)D_h(x^{k_i}, \bar{x}) \right] \\
\leq \Psi(\bar{x}) + \lim_{i \to \infty} (\lambda^{-1} + \mu) \left[ D_h(\bar{x}, y^{k_i-1}) + D_h(x^{k_i}, \bar{x}) \right],
\]
where the second inequality is based on $L \leq \bar{\lambda}^{-1} \leq \lambda^{-1}$ in BPGe. From (12), together with the continuity of $\nabla h$, we obtain
\[
\lim_{i \to \infty} \left[ D_h(\bar{x}, y^{k_i-1}) + D_h(x^{k_i}, \bar{x}) \right] \\
\leq \lim_{i \to \infty} \left[ D_h(\bar{x}, y^{k_i-1}) + D_h(y^{k_i-1}, \bar{x}) + D_h(x^{k_i}, \bar{x}) + D_h(\bar{x}, x^{k_i}) \right] \\
\leq \lim_{i \to \infty} \left[ \|\nabla h(y^{k_i-1}) - \nabla h(\bar{x})\|\|y^{k_i-1} - \bar{x}\| + \|\nabla h(x^{k_i}) - \nabla h(\bar{x})\|\|x^{k_i} - \bar{x}\| \right] \\
= 0.
\]
Hence we have
\[
\limsup_{i \to \infty} \Psi(x^{k_i}) \leq \Psi(\bar{x}). \tag{20}
\]
Combining (15) and (20) yields $\Psi(x^{k_i}) \to \Psi(\bar{x})$, $k_i \to \infty$.

Thus, according to these results, and the closedness of $\partial \Psi$ (see, Exercise 8 in Page 80 [2]), we have $0 \in \partial \Psi(\bar{x})$.

(iii) In view of Lemma 1 and (i), the sequence $\{H_{k, M}\}$ is convergent and $D_h(x^{k-1}, x^k) \to 0$, these together with the definition of $H_{k, M}$ imply $\lim_{k \to \infty} \Psi(x^k)$ exists, denoted as $\zeta$. According to the last part of the proof in (ii), and taking $\bar{x} \in \omega(x^0)$ with a convergent subsequence $\{x^{k_i}\}$ such that $\lim_{i \to \infty} x^{k_i} = \bar{x}$, we know that
\[
\zeta = \lim_{i \to \infty} \Psi(x^{k_i}) = \Psi(\bar{x}).
\]
Thus the conclusion is completed since $\bar{x}$ is arbitrary. \hfill \Box

Next, we prove a global $O\left(\frac{1}{K}\right)$ sublinear convergence rate for the sequence $\min_{k \in \mathbb{N}} D_h(x^{k-1}, x^k)$ of the algorithm. In fact, the linear convergence rate can also be got if we add more assumptions, like KL property and concrete KL exponent (we refer to [3]), based on similar deductions as in [4] Theorem 6.3).

**Corollary 2.** Suppose $\rho < \lambda/\bar{\lambda}$ and $\{x^k\}_{k \in \mathbb{N}}$ be a sequence generated from $x^0$ by BPGe. Then for all $K \geq 1$, $\min_{1 \leq k \leq K} D_h(x^{k-1}, x^k)$ converges with a sublinear rate as $O\left(\frac{1}{K}\right)$.

**Proof.** Set $M = \frac{1}{\lambda}$, recall (11), now for $K \geq 1$,
\[
0 \leq \sum_{i=1}^{K} \left( \frac{\lambda^{-1} - \rho \lambda^{-1}}{\lambda^{-1}} \right) D_h(x^{k-1}, x^k) \leq H_{1,M} - H_{K+1,M}.
\]
Hence we obtain
\[
\min_{1 \leq k \leq K} D_h(x^{k-1}, x^k) \leq \frac{H_{1,M} - H_{K+1,M}}{K \left( \lambda^{-1} - \rho \lambda^{-1} \right)}.
\] (21)

Next, we focus on performing a global convergence analysis. We aim to prove that the sequence \( \{x^k\}_{k \in \mathbb{N}} \) generated by BPGe converges to a critical point of the objective function \( \Psi \) defined in (P). In order to prove global convergence, we use the proof methodology introduced in reference [36]. This proof methodology proves global convergence result for several types of nonconvex nonsmooth problems. Other similar forms were referred in many works [37, Section 3.2], [38, Section 4], [4, Section 4.2].

For the reader’s convenience, we firstly describe the proof methodology summarized in [37, Theorem 3.7] with a few modifications and then we apply it to prove the convergence of BPGe in Theorem 2.

**Theorem 1.** [37, Theorem 3.7] Let \( F : \mathbb{R}^{2d} \to (-\infty, \infty] \) be a proper lower-semi-continuous function. Assume that \( \{z^k\}_{k \in \mathbb{N}} := \{(x^k, x^{k-1})\}_{k \in \mathbb{N}} \) is a sequence generated by a general algorithm from \( z^0 := (x^0, x^0) \), for which the following three conditions are satisfied for any \( k \in \mathbb{N} \).

(H1) For each \( k \in \mathbb{N} \), there exists a positive \( 'a' \) such that
\[
F(z^{k+1}) + a \|x^k - x^{k-1}\|^2 \leq F(z^k), \quad \forall k \in \mathbb{N}.
\]

(H2) For each \( k \in \mathbb{N} \), there exists a positive \( 'b' \) such that for some \( v^{k+1} \in \partial F(z^{k+1}) \) we have
\[
\|v^{k+1}\| \leq b \left( \|x^{k+1} - x^k\| + \|x^k - x^{k-1}\| \right), \quad \forall k \in \mathbb{N}.
\]

(H3) There exists a subsequence \( \{z^{k_j}\}_{j \in \mathbb{N}} \) such that \( z^{k_j} \to \tilde{z} \) and \( F(z^{k_j}) \to F(\tilde{z}) \).

Moreover, if \( F \) have the Kurdyka-Lojasiewicz property at the limit point \( \tilde{z} = (\tilde{x}, \tilde{x}) \) specified in (H3). Then, the sequence \( \{x^k\}_{k \in \mathbb{N}} \) has finite length, i.e., \( \sum_{k=1}^{\infty} \|x^k - x^{k-1}\| < \infty \), and converges to \( \bar{x} = \tilde{x} \) as \( k \to \infty \), where \( (\bar{x}, \bar{x}) \) is a critical point of \( F \).

In our paper, what we need is to verify that the conditions given in Theorem 1 are satisfied for \( F(x, y) = \Psi(x) + MD_h(y, x) \) and the sequence \( (x^k, x^{k-1})_{k \in \mathbb{N}} \in \mathbb{R}^{2d} \) generated by the BPGe algorithm.

In order to guarantee the three conditions hold, we need another assumption. The first two requirements of the assumption were also required in [4, see Assumption D(ii)], and the third condition is easily verified.

**Assumption 3.** (i) \( h \) is \( \sigma \)-strongly convex on \( \mathbb{R}^d \).

(ii) \( \nabla h, \nabla f \) are Lipschitz continuous on any bounded subset of \( \mathbb{R}^d \).

(iii) There exists a bounded \( u \) such that \( u \in \partial(\nabla h) \) on any bounded subset of \( \mathbb{R}^d \).
In fact, Assumption 3(iii) can guarantee that Assumption 1(ii) hold for the bounded sequence \( \{x_k\}_{k \in \mathbb{N}} \).

The next task is to verify the three conditions one by one. Then, together with Theorem 1, we obtain the result that, under proper parameter selection, the whole sequence generated by BPGe converges to a critical point of the objective function.

**Theorem 2.** Suppose \( \rho < \frac{1}{2}\lambda \). Let \( \{x^k\}_{k \in \mathbb{N}} \) be a sequence generated from \( x^0 \) by BPGe. If \( F(x, y) = \Psi(x) + MD_h(y, x) \) (where \( M \in (\rho \lambda^{-1}, \frac{1}{\lambda}] \)) satisfies the Kurdyka–Lojasiewicz property at some limit point \( \bar{z} = (\bar{x}, \bar{x}) \in \mathbb{R}^{2d} \) and Assumption 3 holds, then

(i) The sequence \( \{x^k\}_{k \in \mathbb{N}} \) has finite length, i.e., \( \sum_{k=1}^{\infty} \|x^k - x^{k-1}\| < \infty \).

(ii) \( x^k \to \bar{x} \) as \( k \to \infty \), and \( \bar{x} \) is a critical point of \( \Psi \).

**Proof.** We first verify the three conditions of the Theorem 1 for function \( H \) and BPGe algorithm.

(H1) According to Assumption 3, since \( h \) is strongly convex, assume that \( h \) is \( \sigma \)-strongly convex, that is \( D_h(x, y) \geq \frac{\sigma}{2} \|x - y\|^2 \) for any \( x, y \in \mathbb{R}^d \). We denote \( a = \frac{\sigma}{2} (M - \rho \lambda^{-1}) \). For any \( k \in \mathbb{N} \),

\[
F(x^{k+1}, x^k) + a\|x^k - x^{k-1}\|^2 \leq F(x^{k+1}, x^k) + (M - \rho \lambda^{-1})D_h(x^{k-1}, x^k) \\
\leq F(x^{k+1}, x^k) + (M - \rho \lambda^{-1})D_h(x^{k-1}, x^k) \\
= H_{k+1, M} + (M - \rho \lambda^{-1})D_h(x^{k-1}, x^k) \\
\leq H_{k, M} + (M - \lambda^{-1}) D_h(x^k, x^{k+1}) \\
\leq H_{k, M} = F(x^k, x^{k-1}),
\]

where the first inequality is based on the strongly convexity of \( h \), the second inequality is based on \( \lambda \leq \lambda_k \), the third and the last equality is from the definitions of \( H_{k, M} \) and \( F \), the fourth inequality is from Lemma 1(ii), and the fifth inequality is according to the nonnegativeness of \( (M - \lambda^{-1}) D_h(x^k, x^{k+1}) \). Thus (H1) is verified.

(H2) From the optimal condition 2, there exists \( -\nabla f(y^k) + \lambda^{-1}_k (\nabla h(y^k) - \nabla h(x^{k+1})) \in \partial g(x^{k+1}) \). Due to Corollary 1, \( \{x^k\}_{k \in \mathbb{N}} \) generated by BPGe is bounded, and so also \( \{y^k\}_{k \in \mathbb{N}} \) is bounded. Thus, according to Assumption 3(iii), there exists a bounded \( u_k \in \partial \nabla h(x^k) \), and

\[
v_{k+1} = (\nabla f(x^{k+1}) - \nabla f(y^k) - \lambda^{-1}_k (\nabla h(x^{k+1}) - \nabla h(y^k)) - M(u_k, x^{k+1} - x^k), M(\nabla h(x^k) - \nabla h(x^{k+1})) \),
\]

such that \( v_{k+1} \in \partial F(x^{k+1}, x^k) \). According to Assumption 3, there exist \( L_f, L_h, \delta \) such that for any \( k \in \mathbb{N} \), \( \|\nabla h(x^{k+1}) - \nabla h(y^k)\| \leq L_h \|x^{k+1} - y^k\|, \|\nabla f(x^{k+1}) - \nabla f(y^k)\| \leq L_f \|x^{k+1} - y^k\|, \|u_k\| \leq \delta \). Hence

\[
\|v_{k+1}\| \leq (L_f + \lambda^{-1}_k L_h) \|x^{k+1} - y^k\| + M(\delta + L_h) \|x^{k+1} - x^k\| \\
\leq (L_f + (\lambda^{-1}_k + M)L_h + M\delta) \|x^{k+1} - x^k\| + (L_f + \lambda^{-1}_k L_h) \|x^k - x^{k-1}\| \\
\leq (L_f + (\lambda^{-1}_k + M)L_h + M\delta) (\|x^{k+1} - x^k\| + \|x^k - x^{k-1}\|),
\]

And so, (H2) is satisfied.

(H3) Condition (H3) naturally follows from Lemma 2(ii).
According to Theorem [1], combining the three conditions given in Theorem [1] and KL property at \( \tilde{z} \) could guarantee that conclusion (i) holds. Conclusion (ii) is followed by Theorem [2(i)]. Thus \( \{x^k\}_{k \in \mathbb{N}} \) is a Cauchy sequence of \( \mathbb{R}^d \) and converges to its limit point \( \tilde{x} \). From Theorem [1], \( \tilde{x} \) is the critical point.

5 Numerical Results

In this section we perform several numerical tests in order to show the behaviour and the convergence speed up obtained when using the BPGe algorithm. We consider two important optimization problems in which the differentiable part of the objective does not admit a global Lipschitz continuous gradient: a convex Poisson linear inverse problem and a nonconvex quadratic inverse problem (and so the PG and PGe algorithms cannot be applied to these problems). It is important to remark that for cases where the differentiable part of the objective admits a global Lipschitz continuous gradient the BPG and BPGe algorithms become the PG and PGe algorithms, respectively. That is, the BPG and BPGe methods can be applied but the performance in these cases it was already shown in [8].

The main parameters in BPGe algorithm are the stepsizes \( \lambda_k \) in Algorithm 1, and the parameter \( \rho \) that gives the extrapolation coefficients \( \beta_k \) in the line search method of Algorithm 2. In our tests we consider fixed stepsizes \( \lambda_k = \lambda \). The influence of both parameters \( \{\lambda, \rho\} \) in order to fix suitable values is studied below in the tests.

All the numerical experiments have been performed in Matlab 2013a on a PC Intel(R) Xeon(R) CPU E5-2697 (2.6 GHz).

5.1 Application to Poisson Linear Inverse Problems (PLIP)

Poisson Linear Inverse Problems (that is, linear inverse problems in presence of Poisson noise) emerged in many fields, like astronomy, nuclear medicine (e.g., Positron Emission Tomography), inverse problems in fluorescence microscopy [3, 23, 39]. Therefore, the design of methods and estimators for such problems has been studied intensively over the last two decades (for a review, see [23 39]). Often these problems can be represented as a minimization problem like

\[
\min \left\{ d(b, Ax) + \theta g(x) : x \in \mathbb{R}^d_+ \right\} \quad \text{(PLIP)}
\]

where \( \theta > 0 \) is used to weigh matching the data fidelity criteria and its regularizer \( g \), and \( d(\cdot, \cdot) \) denotes a convex proximity measure between two vectors.

A very well-known measure of proximity of two nonnegative vectors \( Ax \) and \( b \) is based on the Kullback-Liebler divergence:

\[
d(b, Ax) := \sum_{i=1}^m \left\{ b_i \log \frac{b_i}{(Ax)_i} + (Ax)_i - b_i \right\}.
\]

which corresponds to noise of the negative Poisson log-likelihood function. It is easy to find that \( f := d(b, Ax) \) has no globally Lipschitz continuous gradient [4], but satisfies \( L\)-smad condition with
Figure 1: Poisson Linear Inverse Problems tests (overdetermined case $m > d$): evolution of the difference $\|\Psi(x_k) - \Psi(x^*)\|$ vs. iteration number, changing the parameters $\{\lambda, \rho\}$ and for several problem sizes (measurements $m$) with fixed vector dimension $d = 100$. 
Figure 2: Poisson Linear Inverse Problems tests (overdetermined case $m > d$): evolution of the objective function $\Psi(x_k)$ vs. iteration number, using the parameter values $\{\lambda = 1/L, \rho = 0.99\}$ and for several problem sizes (measurements $m$) with fixed vector dimension $d = 100$.

a kernel generating distance called Burg's entropy, denoted as

$$h(x) = -\sum_{j=1}^{d} \log x_j, \text{dom } h = \mathbb{R}^d_+,$$

and so now the Bregman distance is given by

$$D_h(x, y) = \sum_{j=1}^{d} \left\{ \frac{x_j}{y_j} - \log \left( \frac{x_j}{y_j} \right) - 1 \right\}.$$

Therefore, we have that

(i) $(f, h)$ is $L$-smad, where $L \geq ||b||_1$ (according to Lemma 7 in [3]), and $f$ is 0-relative weakly convex to $h$ since $f$ is convex;

(ii) Assumptions 1 and 2 hold, but Assumption 3 does not hold.

So, from the convergence Section 4, we can solve this problem using the BPGe algorithm and it is guaranteed that any limit point of the sequence generated by BPGe is a stationary point of the objective function $\Psi$.

An important point in any iterative method is to define suitable error control techniques. As discussed in Section 3, EXIT conditions of the experiments are set when iterations exceed 5000 times or $||x^k - x^{k-1}||/\max\{1, ||x^k||\} \leq 10^{-6}$ (as in [8]).

In the tests, the entries of $A \in \mathbb{R}_{+}^{m \times d}$ and $x \in \mathbb{R}^d_+$ are generated following independent uniform distribution over the interval $[0, 1]$. We consider the case $g(x) \equiv 0$, i.e., we solve the inverse problem without regularization, so now the minimization problem is the standard Poisson type maximum likelihood estimation problem (modulo change of sign to pass to a minimization problem).

As these methods (BPG and BPGe) can be applied to both, overdetermined ($m > d$) and underdetermined ($m < d$) problems, we have performed numerical tests on both cases. First, we present the results obtained in the overdetermined case. As commented before, the main parameters in
**Overdetermined case**

\[
\lambda = \frac{1}{L} \quad \text{and} \quad \lambda = \frac{1}{(3L)}
\]

| m   | d  | \(T_{BPGe}\) | \(T_{BPG}\) | \(N_{BPGe}\) | \(N_{BPG}\) | \(T_{BPGe}\) | \(T_{BPG}\) | \(N_{BPGe}\) | \(N_{BPG}\) |
|-----|----|--------------|--------------|--------------|-------------|--------------|--------------|--------------|-------------|
| 1000| 10 | 0.08         | 0.07         | 74           | 0.07        | 0.21         | 0.22         | 279          | 0.21        |
|     | 50 | 0.40         | 0.15         | 336          | 0.15        | 0.14         | 0.16         | 155          | 0.15\(^a\) |
|     | 100| 1.13         | 0.41         | 574          | 0.40        | 0.32         | 0.10         | 187          | 0.09\(^a\) |
|     | 200| 1.68         | 0.63         | 665          | 0.49        | 0.44         | 0.07         | 226          | 0.07        |
| 5000| 10 | 0.77         | 0.24         | 605          | 0.23        | 0.83         | 0.22         | 745          | 0.21        |
|     | 50 | 3.32         | 0.26         | 1291         | 0.26        | 4.16         | 0.34         | 1353         | 0.32        |
|     | 100| 7.50         | 0.53         | 2602         | 0.52        | 13.97        | 0.96         | 4460         | 0.89        |
|     | 200| 13.43        | 0.72         | 3577         | 0.72        | 20.26        | 1.12         | 5000         | 1.00        |
| 10000|10  | 2.53         | 0.18         | 699          | 0.17        | 0.50         | 0.03         | 141          | 0.03\(^a\) |
|     | 50 | 6.68         | 0.33         | 1543         | 0.31        | 15.36        | 0.68         | 3255         | 0.65        |
|     | 100| 16.75        | 0.70         | 3441         | 0.69        | 23.90        | 1.02         | 5000         | 1.00        |
|     | 200| 30.32        | 0.99         | 4770         | 0.95        | 30.20        | 1.05         | 5000         | 1.00        |

**Underdetermined case**

\[
\lambda = \frac{1}{L} \quad \text{and} \quad \lambda = \frac{1}{(3L)}
\]

| m   | d  | \(T_{BPGe}\) | \(T_{BPG}\) | \(N_{BPGe}\) | \(N_{BPG}\) | \(T_{BPGe}\) | \(T_{BPG}\) | \(N_{BPGe}\) | \(N_{BPG}\) |
|-----|----|--------------|--------------|--------------|-------------|--------------|--------------|--------------|-------------|
| 100 | 1000| 0.60         | 0.15         | 369          | 0.14        | 2.19         | 0.25         | 1314         | 0.26        |
|     | 200 | 5.03         | 0.89         | 1754         | 0.67        | 3.56         | 0.29         | 1298         | 0.26        |
|     | 300 | 4.50         | 0.78         | 1760         | 0.67        | 2.81         | 0.26         | 1315         | 0.26        |
|     | 5000| 9.17         | 0.23         | 1085         | 0.22        | 70.00        | 1.49         | 5000         | 1.00        |
| 700 | 200 | 12.85        | 0.28         | 1378         | 0.28        | 115.16       | 1.34         | 5000         | 1.00        |
| 1000| 345.13 | 1.18         | 5000         | 1.00        |
| 10000| 10000| 210.06       | 0.66         | 3284         | 0.66        | 549.52       | 1.03         | 5000         | 1.00        |
| 2000| 643.71 | 0.89         | 4271         | 0.85        | 886.94       | 1.07         | 5000         | 1.00        |
| 3000| 967.90 | 1.02         | 5000         | 1.00        | 1084.82      | 1.04         | 5000         | 1.00        |

Table 1: Poisson Linear Inverse Problems tests: CPU-time and number of iterations for different cases of \(m\) (number of data) and \(d\) (dimension) for two different values of the \(\lambda\) parameter for **overdetermined** (top) and **underdetermined** (bottom) cases. \(T_{BPGe}\) and \(T_{BPG}\) denote the CPU-time of BPGe and BPG algorithms, and \(N_{BPGe}\) and \(N_{BPG}\) the number of iterations to reach the EXIT criteria. Superscript \(-a\) points out discordant cases related with a fast linear convergence.
Figure 3: Poisson Linear Inverse Problems tests (overdetermined case $m > d$): evolution of the difference $\|\Psi(x_k) - \Psi(x^*)\|$ vs. iteration number, using the parameter values $\{\lambda = 1/L, \rho = 0.99\}$ and for several problem sizes (measurements $m$ and vector dimensions $d$).

BPGe algorithm are the stepsize $\lambda$ and the parameter $\rho$. In order to study briefly the most suitable set of parameters, we analyze the influence of both parameters $\{\lambda, \rho\}$ in Figure 1. In all the pictures we show the evolution of $\|\Psi(x_k) - \Psi(x^*)\|$ (being $x^*$ the approximate solution obtained at termination of each respective algorithm) with respect to the iteration number $k$. With this figure we can study the influence of the parameters with respect to the size of the problem (measurements $m$) with fixed dimension $d = 100$. Globally, we observe that the value $\rho = 0.99$ has the best results, even if for some cases, the set of initial conditions gives rise to a very fast convergence (as in the cases of using $\lambda = 1/(2L)$ for $m = 5000$ and $\rho = 0.95$, where we have a fast linear convergence instead of sublinear). Note that this kind of differences can be observed on other situations, but the average behaviour tells us that the best performance is when we take $\rho = 0.99$. On the other hand, similar comments can be said with respect to the stepsize parameter $\lambda$. The general situation also recommends us to take the highest value $\lambda = 1/L$ (also for both algorithms BPGe and BPG).

In Figure 2, now with the fixed parameter values $\{\lambda = 1/L, \rho = 0.99\}$ and for the overdetermined ($m > d$) case, we show the evolution of the objective function $\Psi(x_k)$ vs. iteration number and for several problem sizes (measurements $m$) with fixed vector dimension $d = 100$. We observe that always the BPGe algorithm is much faster than the BPG one. In order to observe more clearly the faster convergence, we present in Figure 3 much more simulations but now showing the evolution
Figure 4: Poisson Linear Inverse Problems tests (underdetermined case $m < d$): evolution of the difference $\|\Psi(x_k) - \Psi(x^*)\|$ vs. iteration number, changing the parameters $\{\lambda, \rho\}$ and for several problem sizes (measurements $m$) with fixed vector dimension $d = 5000$. 
Figure 5: Poisson Linear Inverse Problems tests (underdetermined case $m < d$): evolution of the difference $\|\Psi(x_k) - \Psi(x^*)\|$ vs. iteration number, using the parameter values $\{\lambda = 1/L, \rho = 0.99\}$ and for several problem sizes (measurements $m$ and vector dimensions $d$).

of $\|\Psi(x_k) - \Psi(x^*)\|$. We note that the differences of both methods are bigger for low dimension $d$ problems, in fact for the most overdetermined problems $m \gg d$.

In the underdetermined case we also analyze the influence of both parameters $\{\lambda, \rho\}$ in Figure 4 with respect to the size of the problem (measurements $m$) with fixed dimension $d = 5000$. Now, we observe that the value of the parameter $\rho$ seems to not affect too much on the global performance of the method, so we will take the value $\rho = 0.99$ when we fix the parameter. On the other hand, similar comments as in the overdetermined case can be said with respect to the stepsize parameter $\lambda$. Now the behaviour is quite regular, and no cases of very fast convergence have been observed, and the fastest convergence is obtained for the highest value $\lambda = 1/L$ (also for both algorithms BPGe and BPG). Therefore, in the rest of tests on this paper we fix the parameter values $\{\lambda = 1/L, \rho = 0.99\}$.

In Figure 5 now with the fixed parameter values $\{\lambda = 1/L, \rho = 0.99\}$ and for the underdetermined
(m < d) case, we observe that always the BPGe algorithm is much faster than the BPG one. But, similarly as in the overdetermined case, the differences are bigger when we use the methods for larger ratios d/m, that is, for the most underdetermined problems m ≪ d.

Finally, in Table 1 we give the CPU-time and number of iterations for different sizes of problems (number of data m and dimension d) for two values of the λ parameter (λ = 1/L and 1/(3L)) for overdetermined (top) and underdetermined (bottom) cases. From the simulations we observe that when the problem has not a very big size (probably because in these other cases longer simulations are needed) the ratios among both methods provide an interesting speed-up, and in most cases the EXIT strategy stops the BPGe algorithm before the maximum number of iterations is reached. On the other hand, we observe that the CPU-time and iteration number ratios are quite similar, and so there are little differences between them. Note that the BPGe algorithm has an extra step, the line search method of Algorithm 2, but it increments quite a few the final CPU-time. On the table we have remarked three discordant cases (superscript –a–) related with a fast linear convergence, instead of sublinear. This is illustrated, for example, on the left bottom plot of Figure 1 (ρ = 0.99, m = 1000) where the green curve, corresponding to λ = 1/(3L) converges faster than the other colours (as it also occurs in other plots of the same figure). Note that for an overdetermined problem with random data some initial conditions and data may be led to a faster convergence. For the underdetermined problem there is a regular behaviour in all the simulations. Therefore, in the Poisson Linear Inverse Problems tests the BPGe algorithm presents a faster performance compared with the BPG algorithm, giving an interesting option for real problems.

5.2 Application to Quadratic Inverse Problems

In the second test (taken from [4]) we show that BPGe algorithm can deal with a nonconvex Quadratic Inverse Problem (QIP) in which the differentiable term has no globally gradient Lipschitz continuous property. This problem is a natural extension of the linear inverse problem, but now using quadratic measurements. It appears in many popular applications, such as signal recovery [10] and phase retrieve [11] from the knowledge of the amplitude of complex signals.

A general description of the Quadratic Inverse Problem is to find the vector $x \in \mathbb{R}^d$ that solves the system

$$x^T A_i x \simeq b_i, \quad i = 1, \ldots, m$$

being $\{A_i \in \mathbb{R}^{d \times d} | i = 1, \ldots, m\}$ a set of symmetric matrices that describes the model, and $b = (b_1, \ldots, b_m) \in \mathbb{R}^m$ a vector of usually noisy measurements.

Following the formalism given in [4, section 5.1], this problem can be formulated as a nonconvex minimization problem as:

$$\min \left\{ \Psi(x) := \frac{1}{4} \sum_{i=1}^{m} (x^T A_i x - b_i)^2 + \theta g(x) : x \in \mathbb{R}^d \right\}, \quad (QIP)$$

where $\theta > 0$ is used to weigh matching the data fidelity criteria and its regularizer $g$. In our experiments, we take a convex $l_1$-norm regularization function $g(x) = \|x\|_1$. Note that the first function $f(x)$ is a nonconvex differentiable function but that does not admit a global Lipschitz continuous gradient.
The main quality of the BPG and BPGe algorithms (as noted to the BPG in [4]) is that these methods can solve the broad class of problems (QIP). To apply BPG and BPGe on the QIP model properly, we first need to identify a suitable function $h$ (Definition 1). In [4], a proper choice has been given as:

$$h(x) = \frac{1}{4} \|x\|_4^4 + \frac{1}{2} \|x\|_2^2,$$

and so now the Bregman distance is given by

$$D_h(x,y) = \{h(x) - h(y) - (\|y\|^2 y + y)^T (x - y)\}.$$

When $L$ is chosen such that $L \geq \sum_{i=1}^m (3\|A_i\|^2 + \|A_i\||b_i|)$ then by [4, Lemma 5.1], $L$-smad condition (Definition 2) holds for the selected functions $f(x), g(x)$ and $h(x)$. Besides, according to the same analysis in [4, Lemma 5.1], we could derive the relative weakly convex parameter as $\mu \geq \sum_{i=1}^m \|A_i\||b_i|$. In conclusion, we have that:

(i) $(f,h)$ is $L$-smad, $f$ is $\mu$-relative weakly convex to $h$.

(ii) Assumptions 1 and 2 are easily verified.

(iii) $f,g,D_h$ are all semi-algebraic, (see for example [34]). One can show inductively that $H_M(x,y) = \Psi(x) + MD_h(x,y)$ is semi-algebraic, thus it has KL property (Definition 4) at any point $(x,x)$. Besides, we could verify that Assumption 3 holds.

It means, from the convergence Section 4 that the sequences generated by BPGe algorithm converge to a critical point of the objective function $\Psi$.

Here, we perform several numerical tests to compare the behaviour of the BPGe and BPG algorithms. As we did with the previous problem (PLIP), we have designed two main families of experiments, considering overdetermined ($m > d$) and underdetermined ($m < d$) cases. To that goal we set different values of $m$ and $d$, and we generate $m$ random rank-1 matrices $A_i = a_i a_i^T$ in $\mathbb{R}^{d \times d}$, where the entries of the vectors $a_i$ are generated following independent Gaussian distributions with zero mean and unit variance. The accurate $x^* := \arg \min \{\Psi(x) : x \in \mathbb{R}^d\}$ is chosen as a sparse vector (the sparsity is 5%) and $b_i = x^T A_i x^*$, $i = 1, \ldots, m$. We set the weight parameter $\theta = 1$ as default.

As a first performance comparison, in Table 2 we give the CPU-time and number of iterations for different sizes of problems (number of data $m$ and dimension $d$) for two values of the $\lambda$ parameter ($\lambda = 1/L$ and $1/(3L)$) for overdetermined case. The values $T_{BPGe}$ and $T_{BPG}$ denote the CPU-time of BPGe and BPG algorithms, and $N_{BPGe}$ and $N_{BPG}$ the number of iterations to reach the EXIT criteria, respectively. From the simulations we observe that the ratios among both methods provide an interesting speed-up, and the EXIT strategy stops the BPGe algorithm before the maximum number of iterations ($k_{max} = 5000$ in this case) is reached. On the other hand, we observe that the CPU-time and iteration number ratios are quite similar, and so there are little differences between them. Therefore, we note again that although the BPGe algorithm has an extra step (the line search method of Algorithm 2), it increments quite a few the final CPU-time. Also, from the data we observe that although the ratio for the BPGe and BPG algorithms for $\lambda = 1/(3L)$ is quite good, the option BPGe with $\lambda = 1/L$ performs many fewer iterations, and so it is the recommended option.
Figure 6: Quadratic Inverse Problems tests (overdetermined case $m > d$) and (underdetermined case $m < d$): evolution of the difference $\|\Psi(x_k) - \Psi(x^*)\|$ vs. iteration number, using the parameter values $\{\lambda = 1/L, \rho = 0.99\}$ and for several problem sizes (measurements $m$ and vector dimensions $d$) and evolution of the objective function $\Psi(x_k)$. 
\[ \lambda = \frac{1}{L} \]

| \( m \) | \( d \) | \( T_{BPG} \) | \( T_{BPG} \) | \( N_{BPG} \) | \( N_{BPG} \) | \( T_{BPG} \) | \( T_{BPG} \) | \( N_{BPG} \) | \( N_{BPG} \) |
|--------|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 10000  | 10     | 0.29 0.53  | 146 0.35   | 0.48 0.28   | 248 0.20   |
|        | 50     | 0.57 0.14  | 271 0.14   | 4.41 0.10   | 480 0.10   |
|        | 100    | 1.16 0.10  | 339 0.08   | 8.73 0.19   | 655 0.13   |
|        | 200    | 10.15 0.15 | 608 0.12   | 17.24 0.31  | 1668 0.33  |
| 20000  | 10     | 0.24 0.34  | 143 0.34   | 0.39 0.26   | 304 0.26   |
|        | 50     | 4.09 0.14  | 266 0.14   | 6.80 0.16   | 465 0.09   |
|        | 100    | 1.79 0.09  | 323 0.09   | 9.39 0.16   | 605 0.12   |
|        | 200    | 66.97 0.18 | 602 0.12   | 40.74 0.28  | 1413 0.28  |
| 30000  | 10     | 0.40 0.44  | 145 0.35   | 3.22 0.27   | 231 0.20   |
|        | 50     | 1.48 0.15  | 261 0.15   | 10.42 0.10  | 472 0.10   |
|        | 100    | 32.79 0.09 | 331 0.09   | 15.06 0.12  | 594 0.12   |
|        | 200    | 153.17 0.12| 554 0.11   | 487.62 0.27 | 1341 0.27  |

Table 2: Quadratic Inverse Problems tests: CPU-time and number of iterations for different cases of \( m \) (number of data) and \( d \) (dimension) for two different values of the \( \lambda \) parameter for the overdetermined case. \( T_{BPG} \) and \( T_{BPG} \) denote the CPU-time of BPGe and BPG algorithms, and \( N_{BPGe} \) and \( N_{BPG} \) the number of iterations to reach the EXIT criteria.

In Figure 6 with the fixed parameter values \( \{ \lambda = 1/L, \rho = 0.99 \} \) and for the overdetermined \((m > d)\) and underdetermined \((m < d)\) cases, we show the evolution of \( \| \Psi(x_k) - \Psi(x^*) \| \). In this problem we observe that the performance of the accelerated BPGe algorithm for the overdetermined case is quite good, giving a linear convergence. In the case of underdetermined the behaviour seems to be sublinear, and it needs more iterations to reach the desired value (in these simulations \( k_{max} = 20000 \)). In both cases the BPGe algorithms performs much better than the BPG one. For the underdetermined case we also show the evolution of the objective function \( \Psi(x_k) \) vs. iteration number to see that in this case the objective function takes large values, and therefore, when applying the EXIT strategy the required precision is obtained (a relative error \( < 10^{-6} \)) giving not too small absolute values.

Therefore, again in the Quadratic Inverse Problems tests the BPGe algorithm presents a faster performance compared with the BPG algorithm, giving an interesting option for real problems.

6 Conclusions

This work have joined two powerful methods to solve large-scale minimization problems and we proposed a new accelerated Bregman proximal gradient algorithm (BPGe) useful for nonconvex and nonsmooth minimization problems. On one hand, we have taken the BPG algorithm \([3]\) able to deal with non-globally Lipschitz continuous gradient problems. Firstly defined for the convex case \([3]\) and later extended to the nonconvex case by \([4]\). And on the other hand, the accelerated extrapolation algorithm (used for instance in the PG algorithm \([8]\)). The use of the Bregman distance paradigm permits to enlarge the number of problems to work with, because we do not need the assumption of global Lipschitz gradient continuity. And with the extrapolation technique the convergence of the method is accelerated.

In this paper we have studied the convergence of the new method, and we have proven that any limit point of the sequence generated by BPGe algorithm is a stationary point of the problem by
choosing parameters properly. Besides, assuming Kurdyka-Łojasiewicz property, we have proven the whole sequences generated by BPGe converges to a stationary point.

Finally, we have applied it to two important practical problems that arise in many fundamental applications (and that not satisfy global Lipschitz gradient continuity assumption): Poisson linear inverse problems and quadratic inverse problems, for both, overdetermined and underdetermined cases. In these tests the BPGe algorithm have shown faster convergence results than the BPG algorithm, and so the new BPGe algorithm seems to be an interesting methodology.

References

[1] D. Bertsekas, Convex optimization theory, Athena Scientific, 2009.

[2] J. Borwein, A. S. Lewis, Convex analysis and nonlinear optimization: theory and examples, Springer Science & Business Media, 2010.

[3] H. H. Bauschke, J. Bolte, M. Teboulle, A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications, Mathematics of Operations Research 42 (2) (2017) 330–348.

[4] J. Bolte, S. Sabach, M. Teboulle, Y. Vaisbourd, First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems, SIAM Journal on Optimization 28 (3) (2018) 2131–2151.

[5] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM journal on imaging sciences 2 (1) (2009) 183–202.

[6] Y. Nesterov, A method of solving a convex programming problem with convergence rate $o(1/k^2)$, Soviet Math. Dokl. 27 (1983) 372–376.

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

[8] B. Wen, X. Chen, T. K. Pong, Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems, SIAM Journal on Optimization 27 (1) (2017) 124–145.

[9] D. L. Donoho, Compressed sensing, IEEE Transactions on information theory 52 (4) (2006) 1289–1306.

[10] A. Beck, Y. C. Eldar, Sparsity constrained nonlinear optimization: Optimality conditions and algorithms, SIAM Journal on Optimization 23 (3) (2013) 1480–1509.

[11] D. R. Luke, Phase retrieval, what’s new, SIAG/OPT Views and News 25 (1) (2017) 1–5.

[12] N. Parikh, S. Boyd, et al., Proximal algorithms, Foundations and Trends® in Optimization 1 (3) (2014) 127–239.

[13] M. Schmidt, N. L. Roux, F. R. Bach, Convergence rates of inexact proximal-gradient methods for convex optimization, in: Advances in neural information processing systems, 2011, pp. 1458–1466.
[14] K. Jiang, D. Sun, K.-C. Toh, An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP, SIAM Journal on Optimization 22 (3) (2012) 1042–1064.

[15] L. Xiao, T. Zhang, A proximal stochastic gradient method with progressive variance reduction, SIAM Journal on Optimization 24 (4) (2014) 2057–2075.

[16] A. Nitanda, Stochastic proximal gradient descent with acceleration techniques, in: Advances in Neural Information Processing Systems, 2014, pp. 1574–1582.

[17] A. I. Chen, A. Ozdaglar, A fast distributed proximal-gradient method, in: Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on, IEEE, 2012, pp. 601–608.

[18] N. D. Vanli, M. Gurbuzbalaban, A. Ozdaglar, Global convergence rate of proximal incremental aggregated gradient methods, SIAM Journal on Optimization 28 (2) (2018) 1282–1300.

[19] K.-C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific Journal of optimization 6 (15) (2010) 615–640.

[20] S. Ghadimi, G. Lan, Accelerated gradient methods for nonconvex nonlinear and stochastic programming, Mathematical Programming 156 (1-2) (2016) 59–99.

[21] H. Li, Z. Lin, Accelerated proximal gradient methods for nonconvex programming, in: Advances in neural information processing systems, 2015, pp. 379–387.

[22] Y. Carmon, J. C. Duchi, O. Hinder, A. Sidford, Accelerated methods for nonconvex optimization, SIAM Journal on Optimization 28 (2) (2018) 1751–1772.

[23] M. Bertero, P. Boccacci, G. Desiderà, G. Vicidomini, Image deblurring with Poisson data: from cells to galaxies, Inverse Problems 25 (12) (2009) 123006.

[24] R. I. Bot, E. R. Csetnek, S. C. László, An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions, EURO Journal on Computational Optimization 4 (1) (2016) 3–25.

[25] F. Hanzely, P. Richtarik, L. Xiao, Accelerated bregman proximal gradient methods for relatively smooth convex optimization, arXiv preprint arXiv:1808.03045.

[26] R. T. Rockafellar, Convex analysis, Princeton university press, 2015.

[27] R. T. Rockafellar, R. J.-B. Wets, Variational analysis, Vol. 317, Springer Science & Business Media, 2009.

[28] H. Lu, R. M. Freund, Y. Nesterov, Relatively smooth convex optimization by first-order methods, and applications, SIAM Journal on Optimization 28 (1) (2018) 333–354.

[29] L. M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, USSR Computational Mathematics and Mathematical Physics 7 (3) (1967) 200–217.

[30] G. Chen, M. Teboulle, Convergence analysis of a proximal-like minimization algorithm using Bregman functions, SIAM Journal on Optimization 3 (3) (1993) 538–543.
[31] M. Teboulle, A simplified view of first order methods for optimization, Mathematical Programming (2018) 1–30.

[32] D. Davis, D. Drusvyatskiy, K. J. MacPhee, Stochastic model-based minimization under high-order growth, arXiv preprint arXiv:1807.00255.

[33] E. Nurminskii, The quasigradient method for the solving of the nonlinear programming problems, Cybernetics 9 (1) (1973) 145–150.

[34] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Mathematical Programming 146 (1-2) (2014) 459–494.

[35] G. Li, T. K. Pong, Calculus of the exponent of Kurdyka–Lojasiewicz inequality and its applications to linear convergence of first-order methods, Foundations of Computational Mathematics (2017) 1–34.

[36] H. Attouch, J. Bolte, B. F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods, Mathematical Programming 137 (1-2) (2013) 91–129.

[37] P. Ochs, Y. Chen, T. Brox, T. Pock, iPiano: Inertial proximal algorithm for nonconvex optimization, SIAM Journal on Imaging Sciences 7 (2) (2014) 1388–1419.

[38] T. Pock, S. Sabach, Inertial proximal alternating linearized minimization (iPALM) for non-convex and nonsmooth problems, SIAM Journal on Imaging Sciences 9 (4) (2016) 1756–1787.

[39] T. Hohage, F. Werner, Inverse problems with Poisson data: statistical regularization theory, applications and algorithms, Inverse Problems 32 (9) (2016) 093001.