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

Convergence Analysis of Bregman Proximal Method

Zhenghao Xu ·

the date of receipt and acceptance should be inserted later

1 Preliminaries

1.1 Bregman divergence

Definition 1 (Legendre function) [1] A function $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is called a Legendre function if h is proper, lsc, strictly convex and essentially smooth.

Definition 2 (Bregman divergence) The Bregman divergence $D_h(x, y)$ induced by Legendre function h is defined as

$$D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

for all $x \in \text{dom } h$ and $y \in \text{dom } \nabla h$.

Definition 3 (Relatively smooth) [2] Let h be a Legendre function and f be a function with dom $f \subseteq$ dom h. Function f is said to be L-smooth relative to h if

$$Lh - f \quad \text{is convex on dom } h,$$

or equivalently,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + LD_h(y, x)$$

for all $x \in \text{dom } h$ and $y \in \text{dom } \nabla h$.

Address(es) of author(s) should be given

1.2 Problem setup

We consider the optimization problem(P)

$$\min_{x\in\mathbb{R}^d} F(x) \coloneqq f(x) + \phi(x) \tag{P}$$

which satisfies following assumptions.
Assumption 1

- h is a Legendre function on $\mathbb{R}^d$,
- $f : \mathbb{R}^d \to \mathbb{R}\cup\{\infty\}$ is a convex, proper, lsc function and is L smooth relative to h,
- $\phi : \mathbb{R}^d \to \mathbb{R}\cup\{\infty\}$ is a convex, proper and lsc function,
- For any $\lambda > 0$, $u, v \in \mathbb{R}^d$ and $x \in \text{int dom } h$, the problem

  $$\min_{u\in\mathbb{R}^d} \phi(u) + \langle v, u - x\rangle + \frac{1}{\lambda} D_h(u, x)$$

  has a unique minimizer in dom ∇h,
- The problem (P) has nonempty solution.

We denote $\mathcal{F}_L(\mathbb{R}^d)$ as the class of function tuples satisfying above assumptions, namely

$$\mathcal{F}_L(\mathbb{R}^d) \coloneqq \left\{(f, h, \phi) \;\middle|\; f, h, \phi \text{ satisfy Assumption 1}\right\}.$$

1.3 Method

We use the Bregman Proximal Gradient (BPG) method to solve (P).

Algorithm 1: Bregman Proximal Gradient (BPG)

```
Input: $f$, $\phi$ and $h$ satisfying Assumption 1, $x_0 \in$ int dom $h$, step size $\lambda \in (0, 1/L]$.
Output: $x_N$
for $k = 0, 1, \ldots, N-1$ do
    $x_{k+1} = \arg\min_{u\in\mathbb{R}^d} \phi(u) + \langle\nabla f(x_k), u - x_k\rangle + \frac{1}{\lambda} D_h(u, x_k)$    (BPG)
```

2 Convergence rate through PEP

We first obtain the convergence rate under Assumption 1, where both f and ϕ are convex. Then we extend our result to the case where f is possibly nonconvex.

We obtain the convergence result of BPG method through solving a relaxed version of performance estimation problem (PEP). The upper bound is derived from an arithmetic proof with the aid of computer algebra system. We evaluate the PEP problem numerically to guess the analytical solution to its dual problem. These dual variables then serve as weights for constraint inequalities. Summing all inequalities with weights arithmetically gives the upper convergence bound as desired. Our approach recovers the typical convergence results.

2.1 PEP formulation

To evaluate the worst case performance of BPG method on problem (P) after N iterations, we formulate its performance estimation problem as follow.

$$\begin{aligned} \max \quad & (F(x_N) - F(x_*))/D_h(x_*, x_0) \\ s.t. \quad & (f, h, \phi) \in \mathcal{F}_L(\mathbb{R}^d), \\ & x_* \text{ is a minimizer of } F, \\ & x_1, \ldots, x_N \text{ are generated by BPG method from } x_0. \end{aligned} \tag{PEP-P}$$

Above PEP problem is hard to solve, thus we make simplification and relaxation of it. We follow the steps introduced in [3–5] with some modifications since some of the properties dose not hold in the composite objective case.

The constraints can be divided into four parts: initial condition, optimality condition, functional assumption and iterative relationship. Initial condition refers to the restriction on initial Bregman divergence $D_h(x_*, x_0)$ between the minimizer x_* of (P) and the initial point of BPG algorithm x_0. Optimality condition includes the constraints induced by the optimality of x_*, namely $F(x_*) = \min_{\mathbb{R}^d} F(x)$ and $0 \in \partial F(x_*)$. The iterative relationship describes the restrictions induced by the fact that $\{x_i\}_{i=1}^N$ are generated from BPG method starting at x_0. Functional assumption is the most difficult part to handle for its infinite dimensionality. We combine it with the iterative relationship and replace them with another interpolation condition, which is finite dimensional and relatively easy to proceed.

The initial condition is encoded as $D_h(x_*, x_0)$ in the objective of (PEP-P), which allows us to derive an upper bound for BPG method with respect to $D_h(x_*, x_0)$. Attributed to the homogeneity between f and h in the relatively-smooth condition, we may assume $D_h(x_*, x_0) = 1$ and thus remove it from the objective whenever $\phi(x) \equiv 0$. This argument dose not hold when we have non-trivial nonsmooth term $\phi(x)$ due to the lack of connection between ϕ and h in our assumption. In this case, we proceed to move the initial divergence $D_h(x_*, x_0)$ from the objective to the constraints and replace it with a bounded constraint $D_h(x_*, x_0) \leq R$ for a constant R. We shall see in the following section that this replacement would still give us a valid upper bound relative to $D_h(x_*, x_0)$.

The second part is the optimality condition, to write explicitly,

$$0 \in \partial F(x_*) = \nabla f(x_*) + \partial \phi(x_*),$$

whenever the optimization problem is considered on $\mathbb{R}^d$.

Interpolation condition is the final part. We may translate the functional constraint $(f, h, \phi) \in \mathcal{F}_L$ and the iterates generated from the algorithm we use into a set of discrete constraints that warrants the capability of interpolation. For each feasible tuple of functions (f, h, ϕ) and a series of points $\{x_i\}_{i \in I}$, $I = \{*, 0, 1, \ldots, N\}$, we have following relations between each pair of points:

$$\begin{aligned}
&f(x_i) - f(x_j) + \langle \nabla f(x_i), x_j - x_i \rangle \le 0 && \text{(convexity of } f\text{)},\\
&h(x_i) - h(x_j) + \langle \nabla h(x_i), x_j - x_i \rangle < 0 && \text{(strict convexity of } h\text{)},\\
&\psi(x_i) - \psi(x_j) + \langle \nabla \psi(x_i), x_j - x_i \rangle \le 0 && \text{(convexity of } \psi\text{)},\\
&\phi(x_i) - \phi(x_j) + \langle \phi'(x_i), x_j - x_i \rangle \le 0 && \text{(convexity of } \phi\text{)},
\end{aligned}$$

where $i, j \in I$, $i \neq j$, $\psi \coloneqq \frac{1}{\lambda} h - f$ and $\phi'(x_i) \in \partial\phi(x_i)$. As for the iterates, we derive the relations directly from the optimality condition of x_{k+1} in (BPG):

$$\begin{aligned}
x_{k+1} &= \operatorname*{arg\,min}_{u \in \mathbb{R}^d} \phi(u) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda} D_h(u, x_k)\\
\Longrightarrow 0 &\in \partial\phi(x_{k+1}) + \nabla f(x_k) + \frac{1}{\lambda}(\nabla h(x_{k+1}) - \nabla h(x_k))\\
\Longrightarrow 0 &= \lambda(\phi'(x_{k+1}) + \nabla f(x_k)) + \nabla h(x_{k+1}) - \nabla h(x_k),
\end{aligned}$$

where $\phi'(x_{k+1}) \in \partial\phi(x_{k+1})$.

We define variables for function values and (sub)gradients at these points and rephrase the interpolation condition of (PEP-P).

$$\begin{aligned}
&\begin{cases}
(f, h, \phi) \in \mathcal{F}_L(\mathbb{R}^d),\\
x_1, \ldots, x_N \text{ are generated from BPG method starting at } x_0
\end{cases}\\
\Longleftrightarrow &\begin{cases}
\exists (f, h, \phi) \in \mathcal{F}_L(\mathbb{R}^d),\\
f_i, h_i, \phi_i \in \mathbb{R},\\
g_i, s_i, w_i, x_i \in \mathbb{R}^d,\\
f_i = f(x_i), g_i = \nabla f(x_i),\\
h_i = h(x_i), s_i = \nabla h(x_i),\\
\phi_i = \phi(x_i), w_i \in \partial\phi(x_i), \quad \text{for all } i \in I,\\
f_i - f_j + \langle g_i, x_j - x_i \rangle \le 0,\\
h_i - h_j + \langle s_i, x_j - x_i \rangle < 0,\\
\frac{1}{\lambda} h_i - f_i - (\frac{1}{\lambda} h_j - f_j) + \left\langle \frac{1}{\lambda} s_i - g_i, x_j - x_i \right\rangle \le 0,\\
\phi_i - \phi_j + \langle w_i, x_j - x_i \rangle \le 0, \quad \text{for all } i, j \in I,\ i \neq j,\\
s_{i+1} - s_i + \lambda(g_i + w_{i+1}) = 0, \quad \text{for all } i \in \{0, 1, \ldots, N-1\}.
\end{cases}
\end{aligned} \tag{C1}$$

The first line of (C1) states the interpolation requirement that there is a tuple of function in the class $\mathcal{F}_L$ such that the function value and (sub)gradient meets the variables. This is guaranteed by the convex interpolation conditions introduced in [4] and the corresponding strictly convex version [5].

Theorem 1 **(Convex interpolation, [4])** *Let I be a finite index set, $\{f_i, g_i, x_i\}_{i\in I} \subseteq (\mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d)$. The following statements are equivalent:*

(i) There exists a proper closed convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ such that

$$f_i = f(x_i),\ g_i \in \partial f(x_i), \quad \forall i \in I.$$

(ii) For all $i, j \in I$, we have

$$f_i - f_j + \langle g_i, x_j - x_i \rangle \le 0.$$

Theorem 2 **(Differentiable and strictly convex interpolation, [5])** *Let I be a finite index set, $\{f_i, g_i, x_i\}_{i\in I} \subseteq (\mathbb{R}\times\mathbb{R}^d\times\mathbb{R}^d)$. The following statements are equivalent:*

(i) There exists a strictly convex and differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ such that

$$f_i = f(x_i),\ g_i = \nabla f(x_i), \quad \forall i \in I.$$

(ii) For all $i, j \in I$, we have

$$\begin{cases} f_i - f_j + \langle g_i, x_j - x_i \rangle < 0, & x_i \ne x_j, \\ f_i = f_j,\ g_i = g_j, & x_i = x_j. \end{cases}$$

With these two theorems, we may take away the functions f, h and ϕ themselves, keeping the variables representing the function values and the (sub)gradients only:

$$(\text{C1}) \iff \begin{cases} f_i, h_i, \phi_i \in \mathbb{R}, \\ g_i, s_i, w_i, x_i \in \mathbb{R}^d, \quad \text{for all } i \in I, \\ f_i - f_j + \langle g_i, x_j - x_i \rangle \le 0, \\ h_i - h_j + \langle s_i, x_j - x_i \rangle < 0, \\ \frac{1}{\lambda}h_i - f_i - (\frac{1}{\lambda}h_j - f_j) + \left\langle \frac{1}{\lambda}s_i - g_i, x_j - x_i \right\rangle \le 0, \\ \phi_i - \phi_j + \langle w_i, x_j - x_i \rangle \le 0, \quad \text{for all } i, j \in I,\ i \ne j, \\ s_{i+1} - s_i + \lambda(g_i + w_{i+1}) = 0, \quad \text{for all } i \in \{0, 1, \dots, N-1\}. \end{cases} \tag{C2}$$

Strict inequalities are somewhat difficult to proceed. Here we consider the relaxed version of PEP, in which we do not require h to be strictly convex. Thus we may remove the constraints on h itself, since its convexity is implied by the fact that both f and $Lh - f$ are convex and so does their sum h. We define function class $\overline{\mathcal{F}_L}(\mathbb{R}^d)$ as

$$\overline{\mathcal{F}_L}(\mathbb{R}^d) \coloneqq \left\{(f, h, \phi) \;\middle|\; f,\ \phi \text{ and } Lh - f \text{ are convex}\right\}.$$

With this definition, we consider the relaxed version of PEP for BPG method.

$$\begin{aligned} \max \quad & (F(x_N) - F(x_*))/D_h(x_*, x_0) \\ s.t. \quad & (f, h, \phi) \in \overline{\mathcal{F}_L}(\mathbb{R}^d), \\ & x_* \text{ is a minimizer of } F, \\ & x_1, \dots, x_N \text{ are generated by BPG method from } x_0. \end{aligned} \tag{1}$$

Replace the constraints with explicit expressions with the argument on each of the three parts we get

$$\max_{\substack{f_i,h_i,\phi_i\in\mathbb{R}\\ g_i,s_i,w_i,x_i\in\mathbb{R}^d}} \quad f_N+\phi_N-(f_*+\phi_*)$$

$$s.t. \quad \begin{cases} h_*-h_0-\langle s_0,x_*-x_0\rangle\le R,\\ g_*+w_*=0,\\ f_i-f_j+\langle g_i,x_j-x_i\rangle\le 0,\\ \frac{1}{\lambda}h_i-f_i-(\frac{1}{\lambda}h_j-f_j)+\left\langle\frac{1}{\lambda}s_i-g_i,x_j-x_i\right\rangle\le 0,\\ \phi_i-\phi_j+\langle w_i,x_j-x_i\rangle\le 0,\quad\text{for all } i,j\in I,\\ s_{i+1}-s_i+\lambda(g_i+w_{i+1})=0,\quad\text{for all } 0\le i\le N-1,\end{cases} \tag{r-PEP-P}$$

where $R>0$ is a predetermined constant and $\lambda\in(0,1/L]$.

Above (r-PEP-P) problem is not convex in its constraints for the existence of inner products. Like the approach introduced in [5], we describe these inner products with Gram matrix notation. Let $P_x=\left(x_*\ x_0\ \dots\ x_N\right)$, $P_g=\left(g_*\ g_0\ \dots\ g_N\right)$, $P_s=\left(s_*\ s_0\ \dots\ s_N\right)$, $P_w=\left(w_*\ w_0\ \dots\ w_N\right)$, and matrix $P\in\mathbb{R}^{d\times(4N+8)}$ bonding these variables together as

$$P:=\left(P_x\ P_g\ P_s\ P_w\right).$$

Let matrix $G=P^TP\in\mathbb{R}^{(4N+8)\times(4N+8)}$. We denote it as

$$G=\begin{pmatrix} G^{x,x} & G^{x,g} & G^{x,s} & G^{x,w}\\ G^{g,x} & G^{g,g} & G^{g,s} & G^{g,w}\\ G^{s,x} & G^{s,g} & G^{s,s} & G^{s,w}\\ G^{w,x} & G^{w,g} & G^{w,s} & G^{w,w}\end{pmatrix}=\begin{pmatrix} P_x^TP_x & P_x^TP_g & P_x^TP_s & P_x^TP_w\\ P_g^TP_x & P_g^TP_g & P_g^TP_s & P_g^TP_w\\ P_s^TP_x & P_s^TP_g & P_s^TP_s & P_s^TP_w\\ P_w^TP_x & P_w^TP_g & P_w^TP_s & P_w^TP_w\end{pmatrix}.$$

Thus we have

$$G^{x,x}_{i,j}=\langle x_i,x_j\rangle, G^{x,g}_{i,j}=\langle x_i,g_j\rangle, G^{x,s}_{i,j}=\langle x_i,s_j\rangle, G^{x,w}_{i,j}=\langle x_i,w_j\rangle,$$

and similar for the rest of the sub matrices for any $i,j\in I$.

With this set of notations, we may rewrite the constraints eliminating the individual variables $\{x_i\}$, $\{g_i\}$, $\{s_i\}$, $\{w_i\}$ and transform the inner products into entries of matrix G. In order to recover these variables from the matrix G, it is only required that G is a Gram matrix so that it can be factorized (by Cholesky decomposition, for instance) into matrix P multiplied by its transpose P^T. This is equivalent to the restriction that G is symmetric and positive semi-definite, $G\succeq 0$. In this case the equality constraints between iterates derived from BPG method is not expressible in terms of matrix G directly. We turn this equality into a relaxed form:

$$\langle s_{i+1}-s_i+\lambda(g_i+w_{i+1}),x_j\rangle=0,\quad 0\le i\le N-1, j\in I,$$

which can be restated using the notations of the Gram matrix. We can also add inner products with other variables beside x_j in order to make sure that $s_{i+1}-s_i+\lambda(g_i+w_{i+1})$, but this is impossible and not necessary. It is impossible since in the large scale settings we usually assume d to be very large so that $d \gg (4N+8)$. It is not necessary, on the other hand, for only the inner products with $\{x_i\}_{i\in I}$ appear in other parts of the PEP formulation. Similar argument applies to the optimality constraint, which we turn into

$$\langle g_* + w_*, x_j\rangle = 0, \quad \forall j \in I.$$

Through the above argument we may rephrase the PEP problem as a semidefinite programming (SDP) problem (sdp-PEP-P):

$$\max_{f_i,h_i,\phi_i\in\mathbb{R}} \quad f_N + \phi_N - (f_* + \phi_*)$$

$$s.t. \quad \begin{cases} h_* - h_0 - G^{s,x}_{0,*} + G^{s,x}_{0,0} \le R, \\ f_i - f_j + G^{g,x}_{i,j} - G^{g,x}_{i,i} \le 0, \\ \frac{1}{\lambda}h_i - f_i - (\frac{1}{\lambda}h_j - f_j) + \frac{1}{\lambda}(G^{s,x}_{i,j} - G^{s,x}_{i,i}) - (G^{g,x}_{i,j} - G^{g,x}_{i,i}) \le 0, \\ \phi_i - \phi_j + G^{w,x}_{i,j} - G^{w,x}_{i,i} \le 0, \quad \text{for all } i,j \in I, \\ G^{s,x}_{i+1,j} - G^{s,x}_{i,j} + \lambda(G^{g,x}_{i,j} + G^{w,x}_{i+1,j}) = 0, \quad 0 \le i \le N-1, j \in I, \\ G^{g,x}_{*,j} + G^{w,x}_{*,j} = 0, \quad \forall j \in I, \\ G \succeq 0. \end{cases} \tag{sdp-PEP-P}$$

2.2 Upper bound through duality

Following previous works [3,6,7,5], we try to solve the dual of (sdp-PEP-P) in order to give an upper bound for BPG method applied on relatively smooth composite convex function minimization problem (P), for the dual of a maximization problem is a minimization problem and the weak duality theorem guarantees that the dual solution is no less than the primal problem. Unlike the gradient method (GM) case as in [3,6], the structure of BPG algorithm does not allow us to simply plug in algorithmic equality constraints into inequality constraints in the PEP formulation to get a simple expression which induces a concise dual form where analytical solution of the dual can be explicitly derived with fewer effort. Here we use YALMIP [8] and SDP solver MOSEK [9] to solve (sdp-PEP-P) numerically which gives numerical dual solutions associated with each constraint. These numerical results lead to an analytical guess on the dual optimal solution and thereafter an analytical upper bound, which is verified through arithmetic proof. The bound and its proof is stated in the next theorem.

Theorem 3 (BPG convergence rate) *Let $L > 0$, (f, h, ϕ) are a tuple of function satistying Assumption 1. Then the sequence $\{x_k\}_{k\ge0}$ generated by Algorithm 1 with constant step size $\lambda \in (0, 1/L]$ satisfies for all $N > 0$,*

$$F(x_N) - F(x_*) \le \frac{D_h(x_*, x_0)}{\lambda N}. \tag{2}$$

Proof We make use of the convexity inequalities in (r-PEP-P) and perform weighted sum of these inequalities, where weights are given by the corresponding dual solutions of (sdp-PEP-P).

Denote $f_i = f(x_i)$, $h_i = h(x_i)$, $\phi_i = \phi(x_i)$, $g_i = \nabla f(x_i)$, $s_i = \nabla h(x_i)$, $w_i = \phi'(x_i) \in \partial\phi(x_i)$ for all $i \in I$ in consistent with the notion in (r-PEP-P).

The proof itself is a pure arithmetic check. The nonzero dual solution and corresponding constraint inequalities involved in the proof:

- $c^{(1)}_{N,i} = \frac{i}{N}, (i = 0, \ldots, N-1)$ for convexity of f between x_i and x_{i+1}:

$$f_{i+1} - f_i + \langle g_{i+1}, x_i - x_{i+1} \rangle \le 0,$$

- $c^{(2)}_{N,i} = \frac{1}{N}, (i = 0, \ldots, N)$ for convexity of f between x_i and x_{i+1}:

$$f_i - f_* + \langle g_i, x_* - x_i \rangle \le 0,$$

- $c^{(3)}_{N,i} = \frac{i}{N}, (i = 1, \ldots, N)$ for convexity of $\frac{1}{\lambda}h - f$ between x_i and x_{i-1}:

$$\frac{1}{\lambda}h_{i-1} - f_{i-1} - (\frac{1}{\lambda}h_i - f_i) + \left\langle \frac{1}{\lambda}s_{i-1} - g_{i-1}, x_i - x_{i-1} \right\rangle \le 0,$$

- $c^{(4)}_{N,i} = \frac{i}{N}, (i = 1, \ldots, N-1)$ for convexity of $\frac{1}{\lambda}h - f$ between x_i and x_{i+1}:

$$\frac{1}{\lambda}h_{i+1} - f_{i+1} - (\frac{1}{\lambda}h_i - f_i) + \left\langle \frac{1}{\lambda}s_{i+1} - g_{i+1}, x_i - x_{i+1} \right\rangle \le 0,$$

- $c^{(5)}_{N,N} = \frac{1}{N}$, for convexity of $\frac{1}{\lambda}h - f$ between x_* and x_i:

$$\frac{1}{\lambda}h_N - f_N - (\frac{1}{\lambda}h_* - f_*) + \left\langle \frac{1}{\lambda}s_N - g_N, x_* - x_N \right\rangle \le 0,$$

- $c^{(6)}_{N,i} = \frac{i}{N}, (i = 0, \ldots, N-1)$ for convexity of ϕ between x_i and x_{i+1}:

$$\phi_{i+1} - \phi_i + \langle w_{i+1}, x_i - x_{i+1} \rangle \le 0,$$

- $c^{(7)}_{N,i} = \frac{1}{N}, (i = 1, \ldots, N)$ for convexity of ϕ between x_* and x_i:

$$\phi_i - \phi_* + \langle w_i, x_* - x_i \rangle \le 0.$$

The weighted sum is

$$\begin{aligned}
&\sum_{i=0}^{N-1} \frac{i}{N} \left(f_{i+1} - f_i + \langle g_{i+1}, x_i - x_{i+1} \rangle\right) \\
&+ \sum_{i=0}^{N} \frac{1}{N} \left(f_i - f_* + \langle g_i, x_* - x_i \rangle\right) \\
&+ \sum_{i=1}^{N} \frac{i}{N} \left(\frac{1}{\lambda} h_{i-1} - f_{i-1} - (\frac{1}{\lambda} h_i - f_i) + \left\langle \frac{1}{\lambda} s_{i-1} - g_{i-1}, x_i - x_{i-1} \right\rangle\right) \\
&+ \sum_{i=1}^{N-1} \frac{i}{N} \left(\frac{1}{\lambda} h_{i+1} - f_{i+1} - (\frac{1}{\lambda} h_i - f_i) + \left\langle \frac{1}{\lambda} s_{i+1} - g_{i+1}, x_i - x_{i+1} \right\rangle\right) \\
&+ \frac{1}{N} \left(\frac{1}{\lambda} h_N - f_N - (\frac{1}{\lambda} h_* - f_*) + \left\langle \frac{1}{\lambda} s_N - g_N, x_* - x_N \right\rangle\right) \\
&+ \sum_{i=0}^{N-1} \frac{i}{N} \left(\phi_{i+1} - \phi_i + \langle w_{i+1}, x_i - x_{i+1} \rangle\right) \\
&+ \sum_{i=1}^{N} \frac{1}{N} \left(\phi_i - \phi_* + \langle w_i, x_* - x_i \rangle\right) \leq 0.
\end{aligned}$$

By substituting equation $s_{i+1} - s_i + \lambda(g_i + w_{i+1}) = 0$ for $i = 0, \ldots, N-1$, one can reformulate above weighted sum as

$$\begin{aligned}
&f_N + \phi_N - f_* - \phi_* - \frac{h_* - h_0 - \langle s_0, x_* - x_0 \rangle}{\lambda N} \\
=&F(x_N) - F(x_*) - \frac{D_h(x_*, x_0)}{\lambda N} \leq 0,
\end{aligned} \tag{3}$$

which gives the bound stated in the theorem. □

Above proof is pure arithmetic, which can be checked through computer algebra system like Mathematica with little effort. This convergence result coincides with the one derived in [5] without nonsmooth term ϕ.

References

1. R. Rockafellar. Convex analysis. In *Princeton Landmarks in Mathematics and Physics*, 1970.
2. Heinz H. Bauschke, J. Bolte, and M. Teboulle. A descent lemma beyond lipschitz gradient continuity: First-order methods revisited and applications. *Math. Oper. Res.*, 42:330–348, 2017.
3. Yoel Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145:451–482, 2014.
4. Adrien B. Taylor, J. Hendrickx, and F. Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161:307–345, 2017.

5. R. Dragomir, Adrien B. Taylor, A. d'Aspremont, and J. Bolte. Optimal complexity and certification of bregman first-order methods. *ArXiv*, abs/1911.08510, 2019.
6. Donghwan Kim and J. Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159:81–107, 2016.
7. Adrien B. Taylor, J. Hendrickx, and F. Glineur. Exact worst-case performance of first-order methods for composite convex optimization. *SIAM J. Optim.*, 27:1283–1313, 2017.
8. Johan Löfberg. Yalmip : a toolbox for modeling and optimization in matlab. 2004.
9. E. Andersen and Knud D. Andersen. The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. 2000.