

Fast proximal algorithms for nonsmooth convex optimization

Adam Ouorou

Orange Labs Research, 44 avenue de la République, 92300 Chatillon, France.

ARTICLE INFO

Keywords:

Nesterov accelerated gradient method
proximal methods
nonsmooth optimization
convex programming

ABSTRACT

In the lines of our approach in [15], where we exploit Nesterov fast gradient concept [12] to the Moreau-Yosida regularization of a convex function, we devise new proximal algorithms for nonsmooth convex optimization. These algorithms need no bundling mechanism to update the stability center while preserving the complexity estimates established in [15]. We report some preliminary computational results on some academic test problem to give a first estimate of their performance in relation with the classical proximal bundle algorithm.

1. Introduction

We are interested in minimizing a nonsmooth convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, over a nonempty convex compact subset S of \mathbb{R}^n . We denote by f^* the optimal value of this problem and x^* an optimal solution. Having generated a number of test points $y^i \in S$, $i = 1, \dots$ with the corresponding function values $f(y^i)$ and subgradients $g^i \in \partial f(y^i)$ via an *oracle* (for f) to form the bundle $B : \{(y^i, f(y^i), g^i)\}$, the function

$$\check{f}_B(x) = \max\{f(y^i) + \langle g^i, x - y^i \rangle, i \in B\}, \quad (1)$$

is a piecewise cutting-plane model for f , which underestimates f , i.e. for any $x \in S$, $\check{f}_B(x) \leq f(x)$. We use the shortcut $i \in B$ for $(y^i, f(y^i), g^i) \in B$. Let F_μ be the Moreau-Yosida regularization of f w.r.t. some $\mu > 0$ assumed fixed in the sequel. The function F_μ is given by

$$F_\mu(x) = \min_{z \in S} \left\{ f(z) + \frac{\mu}{2} \|z - x\|^2 \right\}.$$

Minimizing f is equivalent to minimizing F_μ . Exploiting the fact that F_μ is convex and differentiable, it is proposed in [15], to apply the concept of fast gradient method [12, 13] to F_μ for the minimization of f . This results in the following scheme, starting from any $x^0 = y^0 \in \mathbb{R}^n$,

$$\begin{aligned} y^{k+1} &= \arg \min_{x \in S} \left\{ f(x) + \frac{\mu}{2} \|x - x^k\|^2 \right\} = x^k - \frac{1}{\mu} \nabla F_\mu(x^k), \\ x^{k+1} &= y^{k+1} + \alpha_k (y^{k+1} - y^k), \quad \alpha_k = \lambda_{k+1}^{-1} (\lambda_k - 1), \end{aligned} \quad (2)$$

where $\{\lambda_k\}$ is the Nesterov' sequence defined by

$$\lambda_0 = 1, \quad \lambda_{k+1} = \frac{1 + \sqrt{1 + 4\lambda_k^2}}{2}, \quad k \geq 0.$$

This sequence has the following properties

$$\lambda_{k-1}^2 = \lambda_k^2 - \lambda_k, \quad k \geq 1, \quad \lambda_k^2 = \sum_{i=0}^k \lambda_i, \quad \lambda_k \geq \frac{k+2}{2}, \quad k \geq 0.$$

✉ adam.ouorou@orange.com (A. Ouorou)
ORCID(s):

(3)

The above scheme generates a sequence $\{y^k\}$ of approximations to an optimal point of the considered problem, and a second sequence $\{x^k\}$ of *stability centers*, different from the former. It is possible to use another update for x^{k+1} , as proposed by Güler in [5],

$$x^{k+1} = y^{k+1} + \alpha_k (y^{k+1} - y^k) + \beta_k (y^{k+1} - x^k), \quad (4)$$

where

$$\beta_k = \lambda_k \lambda_{k+1}^{-1}. \quad (5)$$

Computing exactly y^{k+1} , the proximal point of the stability center x^k is out of reach in practice. In [15], we compute an approximate solution through a sequence of quadratic subproblems

$$z^j = \arg \min_{x \in S} \left\{ \check{f}_{B_j}(x) + \frac{\mu}{2} \|x - x^k\|^2 \right\}, \quad j = 1, \dots,$$

As the bundle B_j grows, $\{z^j\}_{j \geq 0}$ tends to y^{k+1} . An approximate proximal point of x^k is identified when the condition

$$f(z^j) - \check{f}_{B_j}(z^j) \leq \varepsilon_k, \quad (6)$$

is satisfied for some positive tolerance ε_k , in which case y^{k+1} is set to z^j (we keep the same notation as for the exact proximal point). Then, the next stability center x^{k+1} is updated using this approximation in place of the exact proximal point in (2) or (4). There are two versions of the above outlined algorithm FPBA, that we denote by FPBA1 and FPBA2 (FPBA stands for Fast Proximal Bundle Algorithm). FPBA1 uses the rule in (2) to update the next prox-center x^{k+1} while FPBA2 uses the *momentum term* proposed by Güler with the sequence $\{\beta_k\}$, cf (4). The complexity estimates of the two algorithms are given respectively as

$$f(y^k) - f^* \leq \frac{2\mu \|x^0 - x^*\|^2}{(k+1)^2} + \vartheta_k, \quad k \geq 1, \quad (7)$$

for FPBA1, and

$$f(y^k) - f^* \leq \frac{\mu \|x^0 - x^*\|^2}{(k+1)^2} + \vartheta_k, \quad k \geq 1, \quad (8)$$

for FPBA2, where x^* is any optimal solution and

$$\vartheta_k = \lambda_{k-1}^{-2} \sum_{i=0}^{k-1} \lambda_i^2 \varepsilon_i \quad (9)$$

is the accumulation of errors at step k , see Theorems 3.1 and 3.2 in [15].

In this paper, we first propose a variant of the algorithm FPBA in [15] that uses no bundling mechanism to update the stability center x^k and second, take inspiration from [10, 14] and the recent survey on bundle methods [4] we propose two other proximal algorithms in the lines of our approach. For ease of exposition, we limit our development to the case with $\beta_k = 0$, $k \geq 0$. The results for the other case can be obtained conjointly with the development and arguments used in [15]. We assume that \mathcal{S} is simple enough to allow solving easily all the linear and quadratic subproblems in the paper.

2. Fast proximal cutting plane algorithm

In Section 4 of [15], an analysis of the accumulated errors (9) shows that one can tolerate large errors in early iterations but require smaller and smaller errors in the progress of the algorithms. Based on this, one may be content with only one quadratic subproblem at each step k , obtaining the proximal point of x^k w.r.t to \check{f}_{B_k} , which is an approximation of the exact proximal point of x^k w.r.t. f with error $\varepsilon_k = f(y^{k+1}) - \check{f}_{B_k}(y^{k+1}) \geq 0$. There is no need to distinguish between serious and null steps as in FPBA or classical proximal bundle algorithms. The resulting variant of FPBA, which we term as Fast Proximal Cutting Plane Algorithm (FPCPA), is as follows.

Algorithm 1.

0. Choose $x^0 = y^0 \in \mathbb{R}^n$ and the sequence $\{\beta_k\}_{k \geq 0}$. Set $k = 0$.
1. Compute $f(y^k)$, $g^k \in \partial f(y^k)$ and update B_k .
2. If $g^k = 0$, terminate.
3. Compute

$$y^{k+1} = \arg \min_{x \in \mathcal{S}} \left\{ \check{f}_{B_k}(x) + \frac{\mu}{2} \|x - x^k\|^2 \right\}, \quad (10)$$

$$\text{and } x^{k+1} = y^{k+1} + \alpha_k(y^{k+1} - y^k) + \beta_k(y^{k+1} - x^k).$$

4. Set $k = k + 1$ and go to Step 1.

In this algorithm, the choice of the sequence $\{\beta_k\}$ as $\beta_k = 0$, $k \geq 0$ or $\beta_k = \lambda_k \lambda_{k+1}^{-1}$, $k \geq 0$, results in two versions of the algorithm, which we denote respectively by FPCPA1 and FPCPA2. They preserve respectively the complexity estimates (7) and (8). It is possible to use a proximity parameter that depends on k with the same complexity estimates, provided that $\mu_0 = \mu$ and $\mu_k \leq \mu_{k-1}$, $k \geq 1$, see Proposition 3.1 in [15]. With a dynamic setting of the proximity parameter, Algorithm 1 appears as an implementable version of the inertial proximal algorithm [1]. The convergence of this algorithm may be derived from that of Algorithm 3 below.

3. Fast level algorithm

Define the level l_k by

$$l_k = \kappa f_{\text{low}}^k + (1 - \kappa) f_{\text{best}}^k = f_{\text{best}}^k - \kappa \Delta_k, \quad (11)$$

where

- $0 < \kappa < 1$ is the level parameter,
- f_{best}^k is the best objective value found at step k ,
- f_{low}^k is a finite lower bound on f^* ,
- $\Delta_k = f_{\text{best}}^k - f_{\text{low}}^k \geq 0$.

By interpreting the term $\check{f}_{B_k}(x)$ in (10) as the dualization of a constraint $\check{f}_{B_k}(x) \leq l_k$, an alternative to (10), consists in projecting x^k on the l_k -level set of \check{f}_{B_k} , see [4, 6]. The corresponding algorithm, denoted by FLA (for Fast Level Algorithm) is as follows.

Algorithm 2.

0. Choose $x^0 = y^0 \in \mathbb{R}^n$ and the sequence $\{\beta_k\}_{k \geq 0}$. Set $k = 0$.
1. Compute $f(y^k)$ and $g^k \in \partial f(y^k)$ and update B_k .
2. Update f_{best}^k , f_{low}^k . Set $\Delta_k = f_{\text{best}}^k - f_{\text{low}}^k$ and $l_k = f_{\text{best}}^k - \kappa \Delta_k$.
3. If $\Delta_k \leq \varepsilon$ or $g^k = 0$, stop.
4. Compute

$$y^{k+1} = \arg \min_{x \in \mathcal{S}} \left\{ \frac{1}{2} \|x - x^k\|^2 : \check{f}_{B_k}(x) \leq l_k \right\} \quad (12)$$

$$\text{and } x^{k+1} = y^{k+1} + \alpha_k(y^{k+1} - y^k) + \beta_k(y^{k+1} - x^k).$$

5. Set $k = k + 1$ and loop to Step 1.

The convergence property of this algorithm is given below.

Theorem 1. For the sequence $\{y^k\}$ generated by Algorithm 2 with $\beta_k = 0$, $k \geq 0$, we have

$$f(y^k) - f^* \leq \frac{2\mu \|x^0 - x^*\|^2}{t_0(k+1)^2} + \vartheta_k, \quad k \geq 1,$$

where t_k is the optimal dual solution of (21).

The proof is given after that of the next algorithm.

4. Fast doubly stabilized algorithm

In this section, taking inspiration from [14], we propose an algorithm with the aim to leverage the good features of the two previous ones by combining the quadratic problems (10) and (12) into a single quadratic subproblem as follows

$$\min_{x \in \mathcal{S}} \left\{ \check{f}_{B_k}(x) + \frac{\mu_k}{2} \|x - x^k\|^2 : \check{f}_{B_k}(x) \leq l_k \right\},$$

or equivalently

$$\min_{(x,r) \in \mathcal{S} \times \mathbb{R}} \left\{ r + \frac{\mu_k}{2} \|x - x^k\|^2 : \check{f}_{B_k}(x) \leq r, r \leq l_k \right\}. \quad (13)$$

For a reason to be apparent shortly, here the proximity parameter needs to depend on k . The resulting algorithm is as follows, we term it as Fast Doubly Stabilized Algorithm (FDSA for short), keeping the wording of [14].

Algorithm 3.

0. Choose $x^0 = y^0 \in \mathbb{R}^n$ and the sequence $\{\beta_k\}_{k \geq 0}$. Set $k = 0$.
1. Compute $f(y^k)$ and $g^k \in \partial f(y^k)$ and update \mathcal{B}_k .
2. Update f_{best}^k, f_{low}^k . Set $\Delta_k = f_{best}^k - f_{low}^k$ and $l_k = f_{best}^k - \kappa \Delta_k$.
3. If $\Delta_k \leq \varepsilon$ or $g^k = 0$, stop.
4. Compute the x -solution y^{k+1} of (13) and set $x^{k+1} = y^{k+1} + \alpha_k(y^{k+1} - y^k) + \beta_k(y^{k+1} - x^k)$.
5. Set $k = k + 1$ and loop to Step 1.

Its convergence is given by the next result.

Theorem 2. Given some $\mu > 0$, assume that the sequence $\{\mu_k\}$ satisfies $\mu_0 = \mu$ and $\mu_k t_{k-1} \leq \mu_{k-1} t_k$ for $k \geq 1$. Then, for the sequence $\{y^k\}$ generated by Algorithm 3 with $\beta_k = 0, k \geq 0$, we have

$$f(y^k) - f^* \leq \frac{2\mu \|x^0 - x^*\|^2}{t_0(k+1)^2} + \vartheta_k, \quad k \geq 1,$$

where t_k is the optimal dual solution associated with the constraint $\check{f}_{\mathcal{B}_k}(x) \leq r$ in (13) and ϑ_k is given by (9).

PROOF. The proof uses the arguments of Lemma 3.1, 3.2 and Theorem 3.1. For the paper to be self-contained, we provide a complete proof.

The KKT conditions for (13) imply that there exist $p_f^k \in \partial \check{f}_{\mathcal{B}_k}(y^{k+1})$, $p_S^k \in \partial \mathcal{I}_S(y^{k+1})$ and real numbers $t_k, \tau_k \geq 0$ such that

$$\begin{aligned} \mu_k(y^{k+1} - x^k) + t_k(p_f^k + p_S^k) &= 0, \quad t_k[\check{f}_{\mathcal{B}_k}(y^{k+1}) - r^k] = 0, \\ 1 - t_k + \tau_k &= 0, \quad \tau_k(r^k - l_k) = 0, \end{aligned} \quad (14)$$

where r^k is the r -solution of (13). These conditions imply that $t_k = \tau_k + 1 \geq 1$ and

$$\gamma_k(x^k - y^{k+1}) = p_f^k + p_S^k \quad \text{where } \gamma_k = t_k^{-1} \mu_k (\leq \mu_k). \quad (15)$$

Recall that $\partial \mathcal{I}_S(x)$ is the normal cone of S at x i.e.

$$\partial \mathcal{I}_S(x) = \{y \in \mathbb{R}^n : \langle y, z - x \rangle \leq 0, \quad z \in S\}.$$

We have for any $x \in S$,

$$\begin{aligned} \langle p_f^k + p_S^k, x - y^{k+1} \rangle &= \langle p_f^k, x - y^{k+1} \rangle + \langle p_S^k, x - y^{k+1} \rangle \\ &\leq \langle p_f^k, x - y^{k+1} \rangle. \end{aligned}$$

Therefore, as $p_f^k \in \partial \check{f}_{\mathcal{B}_k}(y^{k+1})$, we get for any $x \in S$,

$$\check{f}_{\mathcal{B}_k}(y^{k+1}) + \langle p_f^k + p_S^k, x - y^{k+1} \rangle \leq \check{f}_{\mathcal{B}_k}(x) \leq f(x).$$

and $f(y^{k+1}) + \langle p_f^k + p_S^k, x - y^{k+1} \rangle - \varepsilon_k \leq f(x)$, with $\varepsilon_k = f(y^{k+1}) - \check{f}_{\mathcal{B}_k}(y^{k+1})$. In other words, $p_f^k + p_S^k \in \partial_{\varepsilon_k} f(y^{k+1})$. Using (15), we have for any $x \in S$,

$$f(x) \geq f(y^{k+1}) + \gamma_k \langle x^k - y^{k+1}, x - y^{k+1} \rangle - \varepsilon_k. \quad (16)$$

Let $\delta_k = f(y^k) - f^*$. Taking $x = y^k (\in S)$ in (16) and multiplying the resulting inequality with $\lambda_k - 1$ give

$$(\lambda_k - 1)(\delta_k - \delta_{k+1}) \geq \gamma_k \langle x^k - y^{k+1}, \lambda_k(y^k - y^{k+1}) + y^{k+1} - y^k \rangle - \lambda_k \varepsilon_k.$$

We add this inequality with the one resulting from (16) with $x = x^* (\in S)$ and get

$$(\lambda_k - 1)\delta_k - \lambda_k \delta_{k+1} \geq \gamma_k \langle x^k - y^{k+1}, \lambda_k(y^k - y^{k+1}) + x^* - y^k \rangle - \lambda_k \varepsilon_k$$

Now, multiplying the above inequality by λ_k and using the first relation in (3) yield

$$\lambda_{k-1}^2 \delta_k - \lambda_k^2 \delta_{k+1} \geq \gamma_k \langle u^k, v^k \rangle - \lambda_k^2 \varepsilon_k, \quad (17)$$

where $u^k = \lambda_k(y^{k+1} - x^k)$ and $v^k = \lambda_k(y^{k+1} - y^k) + y^k - x^*$. For any $u, v \in \mathbb{R}^n$, we have (parallelogram law)

$$\langle u, v \rangle = \frac{1}{2}(\|u\|^2 + \|v\|^2 - \|u - v\|^2) \geq \frac{1}{2}(\|v\|^2 - \|u - v\|^2). \quad (18)$$

Hence

$$\lambda_{k-1}^2 \delta_k - \lambda_k^2 \delta_{k+1} \geq \frac{\gamma_k}{2}(\|v^k\|^2 - \|v^k - u^k\|^2) - \lambda_k^2 \varepsilon_k.$$

Let $w^k = v^k - u^k = \lambda_k(x^k - y^k) + y^k - x^*$, $k \geq 0$. Then,

$$\begin{aligned} w^{k+1} &= \lambda_{k+1}(x^{k+1} - y^{k+1}) + y^{k+1} - x^* \\ &\stackrel{(2)}{=} (\lambda_k - 1)(y^{k+1} - y^k) + y^{k+1} - x^* \\ &= v^k, \end{aligned}$$

and

$$\lambda_{k-1}^2 \delta_k - \lambda_k^2 \delta_{k+1} \geq \frac{\gamma_k}{2} \|w^{k+1}\|^2 - \frac{\gamma_k}{2} \|w^k\|^2 - \lambda_k^2 \varepsilon_k,$$

The assumption $\mu_k t_{k-1} \leq \mu_{k-1} t_k$ implies $\gamma_k \leq \gamma_{k-1}$ and then

$$\lambda_{k-1}^2 \delta_k - \lambda_k^2 \delta_{k+1} \geq \frac{\gamma_k}{2} \|w^{k+1}\|^2 - \frac{\gamma_{k-1}}{2} \|w^k\|^2 - \lambda_k^2 \varepsilon_k.$$

We now sum these inequalities for $i = 1, \dots, k-1$ to get

$$\begin{aligned} \lambda_{k-1}^2 \delta_k &\leq \lambda_0 \delta_1 + \frac{\gamma_0}{2} \|w^1\|^2 + \sum_{i=1}^{k-1} \lambda_i^2 \varepsilon_i - \frac{\gamma_{k-1}}{2} \|w^k\|^2 \\ &\stackrel{\lambda_0=1}{\leq} \delta_1 + \frac{\gamma_0}{2} \|w^1\|^2 + \sum_{i=1}^{k-1} \lambda_i^2 \varepsilon_i. \end{aligned} \quad (19)$$

Using (16) with $x = x^*$ and $k = 0$, we get

$$\begin{aligned} \delta_1 &\leq -\gamma_0 \langle x^0 - y^1, x^* - y^1 \rangle + \varepsilon_0 \\ &\stackrel{(18)}{=} -\frac{\gamma_0}{2} [\|x^0 - y^1\|^2 + \|y^1 - x^*\|^2 - \|x^0 - x^*\|^2] + \varepsilon_0 \\ &\leq -\frac{\gamma_0}{2} \|y^1 - x^*\|^2 + \frac{\gamma_0}{2} \|x^0 - x^*\|^2 + \varepsilon_0 \end{aligned}$$

Since $w^1 = v^0 = \lambda_0(y^1 - y^0) + y^0 - x^* \stackrel{\lambda_0=1}{=} y^1 - x^*$, we have

$$\delta_1 + \frac{\gamma_0}{2} \|w^1\|^2 \leq \frac{\gamma_0}{2} \|x^0 - x^*\|^2 + \varepsilon_0,$$

and from (19),

$$\delta_k \leq \frac{\gamma_0}{2\lambda_{k-1}^2} \|x^0 - x^*\|^2 + \lambda_{k-1}^{-2} \sum_{i=0}^{k-1} \lambda_i^2 \varepsilon_i.$$

It remains to use in the first term of the r.h.s. of this inequality, the fact that $\gamma_0 = t_0^{-1} \mu$ and $\lambda_{k-1} \geq (k+1)/2$ from (3). \square

A few comments are in order.

1. In the same way as Lemma 1 of [14], it can be shown that the x -solution of (13) is either the one of (10) or that of (12). Algorithm 3 makes the choice automatically depending on the value of t_k (in fact this choice depends on the proximity and the level parameters μ_k and κ (defining l_k) which determine t_k). If y_p^{k+1} and y_l^{k+1} denote the respective optimal solutions of the quadratic problems (10) and (12), we have

$$y^{k+1} = \begin{cases} y_p^{k+1} & \text{if } t_k = 1 (\tau_k = 0), \\ y_l^{k+1} & \text{if } t_k > 1 (\tau_k > 0). \end{cases}$$

Because $t_k > 0$, we have $\check{f}_{B_k}(y^{k+1}) = r^k$, $k \geq 0$, while $r^k \leq l_k$ if $t_k = 1$ and $r^k = l_k$ if $t_k > 1$.

2. Discarding the accumulation of errors, the complexity estimate improves slightly compared to the one of Algorithm 1 as $t_0 \geq 1$, cf (7).
3. We get from (16) and Cauchy-Schwartz inequality,

$$f(y^{k+1}) \leq f(x) + \gamma_k \|x^k - y^{k+1}\| \|x - y^{k+1}\| + \varepsilon_k,$$

for any $x \in S$. Therefore, if

$$\gamma_k \|x^k - y^{k+1}\| \leq \varepsilon_1 \quad \text{and} \quad \varepsilon_k \leq \varepsilon_2,$$

for some stopping tolerances $\varepsilon_1, \varepsilon_2 > 0$, then

$$f(y^{k+1}) \leq f(x) + \varepsilon_1 \|x - y^{k+1}\| + \varepsilon_2, \quad \forall x \in S.$$

We can then consider y^{k+1} as an approximate optimal solution if ε_1 and ε_2 are small enough.

4. We can recover the complexity estimate of Algorithm 1 from Theorem 2. Indeed, by replacing (11) with $l_k = +\infty$, we have $\tau_k = 0$, $k \geq 0$ and then, $t_k = 1$ and $\gamma_k = \mu_k$, $k \geq 0$. Algorithm 3 then reduces to Algorithm 1. The complexity estimate given in Theorem 2 becomes (7), the one already given for Algorithm 1, with the assumption that now writes $\mu_0 = \mu$ and $\mu_k \leq \mu_{k-1}$ for $k \geq 1$.
5. For the assumption in Theorem 2 to hold, the proximity parameter needs to depend on k . An example of sequence that satisfies this assumption is given by

$$\mu_0 = \mu \quad \text{and} \quad \mu_k = \gamma_{k-1} = t_{k-1}^{-1} \mu_{k-1}, \quad k \geq 1, \quad (20)$$

due to the fact that $t_k \geq 1$. This rule maintains ($t_{k-1} = 1$) or decreases ($t_{k-1} > 1$) the proximity parameter for the next step. The sequence is only decreasing then

while it may be useful sometimes to increase the proximity parameter. If it was possible to guess t_k , an intuitive choice suggested by the assumption would be $\mu_k = t_k \gamma_{k-1} = t_k t_{k-1}^{-1} \mu_{k-1}$. The ratio $t_k t_{k-1}^{-1}$ would reflect the change between steps $k-1$ and k , maintaining ($t_{k-1} = t_k = 1$), increasing ($t_{k-1} < t_k$) or decreasing ($t_{k-1} > t_k$) the proximity parameter accordingly for the next step. Unfortunately, t_k is obtained only after fixing μ_k and solving (13). \square

In the light of the proof of Theorem 2, we now give that of Theorem 1.

PROOF OF THEOREM 1. It is clear that the unique solution of (12) is also the unique solution of the problem

$$\min_{x \in S} \left\{ \frac{\mu_k}{2} \|x - x^k\|^2 : \check{f}_{B_k}(x) \leq l_k \right\}, \quad (21)$$

for any given $\mu_k > 0$; we set $\mu_0 = \mu$. The KKT conditions for this quadratic problem imply that there exist $p_f^k \in \partial \check{f}_{B_k}(y^{k+1})$, $p_S^k \in \partial \mathcal{I}_S(y^{k+1})$ and $t_k > 0$ (as $x^k \neq y^{k+1}$) such that

$$\mu_k (y^{k+1} - x^k) + t_k (p_f^k + p_S^k) = 0, \quad t_k [\check{f}_{B_k}(y^{k+1}) - l_k] = 0.$$

Therefore, for any $x \in S$,

$$\check{f}_{B_k}(y^{k+1}) + \langle p_f^k + p_S^k, x - y^{k+1} \rangle \leq f(x),$$

or equivalently,

$$f(x) \geq f(y^{k+1}) + \frac{\mu_k}{t_k} \langle x^k - y^{k+1}, x - y^{k+1} \rangle - \varepsilon_k, \quad (22)$$

where $\varepsilon_k = f(y^{k+1}) - \check{f}_{B_k}(y^{k+1})$. The remaining of the proof is similar to that of Theorem 2, under the same assumption $\mu_0 = \mu$, $\mu_k t_{k-1} \leq \mu_{k-1} t_k$, $k \geq 1$. \square

Remark 1. As given in Theorem 1, we cannot state if the complexity estimate Algorithm 2 improves or not over the one of Algorithm 1. Since the sequence $\{y^k\}$ generated by Algorithm 2 is the same as if we use (21) in place of (12) in Step 4, we conjecture that the complexity estimate of Algorithm 2 to be the same as that of Algorithm 3. Indeed, with an appropriate choice of μ_k and κ to have $\tau_k > 0$ for all $k \geq 0$, the sequence $\{y^k\}$ generated by Algorithm 3 is the same as that obtain from Algorithm 2, and then the same complexity estimate as given by Theorem 2. \square

Remark 2. The level parameter κ does not appear explicitly in the complexity estimate of Algorithm 2 as it is for the level bundle algorithms in [10]. In fact it is hidden in t_0 as it influences the level and the dual variables of the level constraints. \square

We now give the complexity estimates of Algorithms 2 and 3 used with the sequence $\{\beta_k\}$ given by (5). The proof is analogue to that of Theorem 3.2 in [15] with the same arguments used in the above proofs of Theorems 1 and 2. The main difference is a better lower bound obtained on the scalar

product $\langle u^k, v^k \rangle$ (cf (17)) thanks to the update of x^{k+1} using a second momentum term proposed by Güler intuitively in [5]. It is shown in [7, 8] by Kim and Fessler that it corresponds to an optimal choice of parameters obtained through a relaxed *performance estimation problem* introduced by Drori and Teboulle to optimize first-order algorithms, see [3].

Theorem 3. *Assume that Algorithms 2 and 3 use the sequence (5) under the assumption of Theorem 2 on the sequence $\{\mu_k\}$. Then, for the sequence $\{y^k\}$ generated, we have*

$$f(y^k) - f^* \leq \frac{\mu \|x^0 - x^*\|^2}{t_0(k+1)^2} + \vartheta_k, \quad k \geq 1,$$

where t_0 is the (respective) dual solution associated with the constraint $\check{f}_{B_k}(x) \leq w$ in respectively the quadratic subproblems (21) and (13) with $\mu_0 = \mu$, and ϑ_k is given by (9).

For the above complexity estimates to be meaningful, it is necessary that the accumulation of errors ϑ_k to not be divergent with the first terms.

Lemma 1. *The sequence $\{\vartheta_k\}$ is bounded above.*

PROOF. Recall that

$$0 \leq \vartheta_k = \sum_{i=0}^{k-1} v_i^k \varepsilon_i \quad \text{where } v_i^k = \lambda_i^2 \lambda_{k-1}^{-2}, \quad i = 0, \dots, k-1.$$

We can observe that $0 \leq v_i^k \leq 1 = v_{k-1}^k$ and the former errors vanish with their weights as they tend to 0 when k grows. We have $\vartheta_{k+1} - \vartheta_k = \varepsilon_k - \lambda_k^{-1} \vartheta_k$, see Section 4 in [15]. Therefore, $\varepsilon_k \leq \lambda_k^{-1} \vartheta_k$ implies $\vartheta_{k+1} \leq \vartheta_k$. From Proposition 4.3 in [2], as the bundle B_k grows, $f(y^{k+1})$ and $\check{f}_{B_k}(y^{k+1})$ get closer to each other i.e. $\varepsilon_k = f(y^{k+1}) - \check{f}_{B_k}(y^{k+1}) \rightarrow 0$ (this means that the last errors vanish as well with high k). We cannot have $\varepsilon_k > \lambda_k^{-1} \vartheta_k (\geq 0)$ for an infinite number of k as it results in the contradiction $0 > 0$. Therefore, there exists some k^* such that $\varepsilon_k \leq \lambda_k^{-1} \vartheta_k$ for $k \geq k^*$ and then $\vartheta_k \leq \vartheta_{k-1} \leq \dots \leq \vartheta_{k^*+1} \leq \vartheta_{k^*} < \infty$, i.e. the sequence $\{\vartheta_k\}_{k \geq k^*}$ is decreasing. \square

Remark 3. We finally observe that in the above development, we may replace \check{f}_k by any other lower model $\underline{f}_k \leq f$ and practical in the sense that the corresponding subproblems analogue to (10), (12) and (13) are easy to solve. In this case, the error at setp k writes $\varepsilon_k = f(y^{k+1}) - \underline{f}_k(y^{k+1}) \geq 0$. \square

5. Numerical experiments

We conducted some preliminary experiments that aim to provide a first look on the performances of the proposed algorithms as compared with the classical proximal bundle algorithm (CPBA). The test problems are the one considered in [15] and described in [11] and the algorithms are implemented using Python 3.5 and Cplex 12.7.1 (with its default

Table 1
Test problems

Problem	Name	n	f^*
1	CB2	2	1.952224
2	CB3	2	2
3	DEM	2	-3
4	QL	2	7.2
5	LQ	2	$-\sqrt{2}$
6	Mifflin1	2	-1
7	Mifflin2	2	-1
8	Rosen-Suzuki	4	-44
9	Shor	5	22.600162
10	Maxquad	10	-0.841408
11	Maxq	20	0
12	Maxl	20	0
13	Goffin	50	0
14	MxHilb	50	0
15	L1Hilb	50	0

settings). FPCPA and CPBA may be run with a fixed proximity parameter, we use here $\mu = 1.0$ which suits for well-scaled problems (FLA does not need the proximity parameter). We ran FDSA with the rule (20). As the sequence $\{\mu_k\}$ is decreasing, we consider a small positive constant μ_{inf} and set

$$\mu_k = \max[\mu_{\text{inf}}, \gamma_{k-1}], \quad k \geq 1, \quad \mu_{\text{inf}} = 10^{-10} \|g^0\|.$$

Our implementation of CPBA uses at each step k a sequence of quadratic subproblems for $j = 1, \dots$

$$z^{k,j} = \arg \min_{(x,r) \in S \times \mathbb{R}} \left\{ \check{f}_{B_{k,j}}(x) + \frac{\mu}{2} \|x - \hat{x}^k\|^2 \right\},$$

where $\hat{x}^0 = x^0$ and $\hat{x}^{k+1} = z^{k,j}$ if

$$f(z^{k,j}) \leq f(\hat{x}^k) - \sigma [f(\hat{x}^k) - \check{f}_{B_{k,j}}(z^{k,j})],$$

in which case we have a descent step, otherwise a null step. In our experiments, we set $\sigma = 0.5$. Since they are non-smooth unconstrained problems, we consider an input parameter $f_{\text{inf}}^{(0)}$ to cope with the assumption of compactness of S . Hence, in Algorithm 2 the lower bound is computed as

$$f_{\text{low}}^k = \min_{(x,r) \in \mathbb{R}^{n+1}} \{ \check{f}_{B_k}(x) : f_{\text{inf}}^{(0)} \leq \check{f}_{B_k}(x) \},$$

and we add the constraint $f_{\text{inf}}^{(0)} \leq \check{f}_{B_k}(x)$ to all the quadratic subproblems to be consistent with our development. There are many tricks to avoid computing f_{low}^k at each step, e.g. [4, 9, 14] but for simplicity, it is updated as indicated above. For all the test problems, we set $\kappa = 0.8$ in (11) and $f_{\text{inf}}^{(0)} = -10$ except for the problems 8 and 9 for which it takes the values -100 and 0 respectively. The maximum number of steps allowed for all the algorithms (number of descent steps in CPBA) is set to 500. With the given optimal functions values, we stop the algorithms on the same basis, when

$$f_{\text{best}}^k - f^* \leq 10^{-6} (1 + |f_{\text{best}}^k|).$$

Table 2
Computational results

Pb	CPBA			FPCPA1	
	#k	#fg	$f - f^*$	#fg	$f - f^*$
1	8	22	9.35E-07	23	1.64E-06
2	7	14	3.51E-07	12	4.54E-08
3	4	7	3.08E-09	8	3.84E-09
4	8	20	2.57E-06	27	1.97E-06
5	4	8	1.29E-07	6	1.75E-06
6	9	27	6.70E-07	20	4.50E-07
7	8	22	1.13E-06	22	3.74E-07
8	9	40	3.90E-05	40	3.74E-05
9	10	43	1.89E-05	43	1.45E-05
10	14	127	3.94E-07	209	9.56E-07
11	78	456	9.59E-07	269	7.60E-07
12	210	231	6.36E-08	81	5.89E-09
13	25	69	2.47E-10	87	8.36E-10
14	500 [†]	504	1.50E-04	264	9.72E-07
15	161	433	9.76E-07	62	8.40E-07

Pb	FLA1		FDSA1	
	#fg	$f - f^*$	#fg	$f - f^*$
1	18	7.46E-07	22	2.53E-06
2	16	4.52E-07	13	3.13E-07
3	11	9.55E-07	11	2.58E-06
4	17	6.60E-06	19	1.47E-06
5	11	2.51E-07	7	3.79E-08
6	21	1.95E-06	16	1.39E-08
7	27	6.73E-07	17	1.91E-06
8	70	2.47E-05	48	9.41E-06
9	59	1.50E-05	41	2.25E-05
10	204	1.34E-06	202	1.29E-06
11	231	9.11E-07	77	2.97E-07
12	48	5.12E-07	8	2.99E-09
13	59	1.86E-10	50	5.81E-12
14	19	2.71E-07	8	1.52E-07
15	26	8.72E-07	8	5.17E-07

[†] maximum number of k -steps (500) reached.

We report on Table 2, the number of steps (column #k) for CPBA, the number of steps is the same as the number of calls to f -oracle (column #fg which also indicates the number of steps of all the algorithms except CPBA) and the absolute difference between f , the best function value found at stop and the optimal value f^* . These experiments show an improvement of the first two proposed algorithms over the classical proximal bundle algorithm since both solve all the test problems within the maximum number of steps allowed to the contrary of the latter. The rule (20) (which gives a decreasing sequence of proximity parameters) seems to be effective with Algorithm 3 which compares favorably to the other algorithms on a majority of the test problems in terms of number of calls to the oracle. We expect further improvement from a more sophisticated management of the proximity parameter in this algorithm.

6. Concluding remarks

We developed new algorithms for nonsmooth convex problems in the line of our previous approach in [15] based of fast gradient methods for smooth optimization. The limited experiments to get a first look at their performances is encouraging. Numerical experiments on large scale problems are needed to confirm these performances including the benefit analysis of the momentum term by Güler. Another

question we would like to investigate is whether the use of non Euclidean entropy-like distances may be beneficial in the present setting as it is for the classical proximal bundle algorithms on certain convex problems. See the recent synthesis in [16] exposing the benefits and limitations of the non Euclidean proximal framework.

Acknowledgements

I'm very thankful to Walid Ben Ameer for his comments on a previous version of the paper.

References

- [1] Attouch, H., Cabot, A., 2018. Convergence rates of inertial forward-backward algorithms. *SIAM J. Optim.* 28(1), 849–874.
- [2] Correa, R., Lemaréchal, C., 1993. New variants of bundle methods. *Mathematical Programming* 62, 261–275.
- [3] Drori, Y., Teboulle, M., 2014. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming* 145, 451–482.
- [4] Frangioni, A., 2020. Standard bundle methods: Untrusted models and duality, in: Bagirov, A., Gaudioso, M., Karmitsa, N., Mäkelä, M. (Eds.), *Numerical nonsmooth optimization*. Springer. chapter 3, pp. 61–116.
- [5] Güler, O., 1992. New proximal point algorithm for convex minimization. *SIAM J. On Optimization* 2(4), 649–664.
- [6] Hiriart-Urruty, J.B., Lemaréchal, C., 1993. *Convex analysis and minimization algorithms*. Springer, Berlin.
- [7] Kim, D., Fessler, J., 2016. Optimized first-order methods for smooth convex minimization. *Mathematical Programming* 159, 81–107.
- [8] Kim, D., Fessler, J., 2018. Generalizing the optimized gradient method for smooth convex minimization. *SIAM J. On Optimization* 28(2), 1920–1950.
- [9] Kiwiel, K., 1995. Proximal level bundle methods for convex non-differentiable optimization, saddle-point problems and variational inequalities. *Mathematical Programming* 69, 89–109.
- [10] Lemaréchal, C., Nemirovskii, A., Nesterov, Y., 1995. New variants of bundle methods. *Mathematical Programming* 69, 111–147.
- [11] Lukšan, L., Vlček, J., 2000. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic.
- [12] Nesterov, Y., 1983. A method for solving a convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR* 269, 543–547.
- [13] Nesterov, Y., 2004. *Introductory lectures on convex programming. A basic course*. Kluwer Boston.
- [14] de Oliveira, W., Solodov, M., 2016. A doubly stabilized bundle method for nonsmooth convex optimization. *Mathematical Programming* 156(1), 125–159.
- [15] Ouorou, A., 2020. Proximal bundle algorithms for nonsmooth convex optimization via fast gradient smooth methods. *arXiv preprint arxiv:2003.03437*.
- [16] Teboulle, M., 2018. A simplified view of first order methods for optimization. *Mathematical Programming* 170, 67–96.