Parameter-free proximal bundle methods with adaptive stepsizes for hybrid convex composite optimization problems

Renato D.C. Monteiro * Honghao Zhang *

October 27, 2023

Abstract

This paper develops a parameter-free adaptive proximal bundle method with two important features: 1) adaptive choice of variable prox stepsizes that "closely fits" the instance under consideration; and 2) adaptive criterion for making the occurrence of serious steps easier. Computational experiments show that our method performs substantially fewer consecutive null steps (i.e., a shorter cycle) while maintaining the number of serious steps under control. As a result, our method performs significantly less number of iterations than its counterparts based on a constant prox stepsize choice and a non-adaptive cycle termination criterion. Moreover, our method is very robust relative to the user-provided initial stepsize.

Key words. hybrid convex composite optimization, iteration-complexity, adaptive stepsize, parameter-free proximal bundle methods.

AMS subject classifications. 49M37, 65K05, 68Q25, 90C25, 90C30, 90C60

1 Introduction

Let $f, h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper lower semi-continuous convex functions such that $\operatorname{dom} h \subseteq \operatorname{dom} f$ and consider the optimization problem

$$\phi_* := \min \{\phi(x) := f(x) + h(x) : x \in \mathbb{R}^n\}. \tag{1}$$

It is said that (1) is a hybrid convex composite optimization (HCCO) problem if there exist nonnegative scalars M, L and a first-order oracle $f' : \operatorname{dom} h \to \mathbb{R}^n$ (i.e., $f'(x) \in \partial f(x)$ for every $x \in \operatorname{dom} h$) satisfying $\|f'(u) - f'(v)\| \leq 2M + L\|u - v\|$ for every $u, v \in \operatorname{dom} h$. The main goal of this paper is to study the complexity of adaptive proximal bundle methods (Ad-GPB) for solving the HCCO problem (1) based on a unified bundle update schemes.

Proximal bundle (PB) methods solve a sequence of prox bundle subproblems

$$x_j = \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \Gamma_j(u) + \frac{1}{2\lambda_j}\|u - x^c\|^2 \right\}, \tag{2}$$

where Γ_j is a bundle approximation of ϕ (i.e., a simple convex function underneath ϕ) and x^c is the current prox center. The prox center is updated to x_j (i.e., a serious step is performed) only when the pair (x_j, λ_j) satisfies a certain error criterion; otherwise, the prox center is kept the same (i.e., a null step is performed). Regardless of the step performed, the bundle Γ_j is updated to account for the newest iterate x_j. In the discussion below, a sequence of consecutive null steps followed by a serious step is referred to as a cycle. Classical PB methods (see e.g. [5, 6, 12, 15, 16, 21, 27]) perform the serious step when x_j satisfies a relaxed descent condition (e.g., see the paragraph containing equation (15) in [18]), which in its unrelaxed form implies that $\phi(x_j) \leq \phi(x^c)$. On the other hand, modern PB methods (see e.g. [8, 18, 19, 20]) perform the

*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. (email: rm88@gatech.edu and hzhang906@gatech.edu). This work was partially supported by AFOSR Grant FA9550-22-1-0088.

serious step when the best ϕ-valued iterate x_j, say y_j, satisfies $\phi(y_j) - m_j \leq \delta$ where m_j is the optimal value of (2) and δ is a suitably chosen tolerance. Although y_j does not necessarily satisfy the descent condition, it does satisfy a δ-relaxed version of it. It is shown in [18, 19] that if $\lambda > 0$ is such that $\max\{\lambda, \lambda^{-1}\} = \mathcal{O}(\varepsilon^{-1})$, then modern PB methods with $\lambda_j = \lambda$ for every j achieve an $\tilde{\mathcal{O}}(\varepsilon^{-2})$ iteration complexity to obtain an ε-solution regardless of whether $\operatorname{dom} h$ is bounded or not. In contrast, papers [5, 12] show that the classical PB methods achieve: i) an $\mathcal{O}(\varepsilon^{-3})$ iteration complexity under the assumption that $\lambda = \Theta(1)$ regardless of whether $\operatorname{dom} h$ is bounded or not; and ii) an $\mathcal{O}(\varepsilon^{-2})$ iteration complexity under the assumption that $\lambda = \Theta(\varepsilon^{-1})$ for the case where $\operatorname{dom} h$ is bounded.

The goal of this paper is to develop a parameter-free adaptive modern PB method, namely Ad-GPB, with two important features: 1) adaptive choice of variable prox stepsizes that "closely fits" the instance under consideration; and 2) adaptive criterion for making the occurrence of serious steps easier. Computational experiments show that Ad-GPB performs substantially fewer consecutive null steps while maintaining the number of serious steps under control. As a result, Ad-GPB performs significantly less number of iterations than the Ad-GPB method of [8, 18, 19]. Moreover, in contrast to GPB, Ad-GPB is very robust with respect to the user-provided initial stepsize.

Several papers (see e.g. [2, 4, 5, 11, 13, 17] of which only [5] deals with complexity analysis), have proposed ways of generating variable prox stepsizes to improve classical PB methods' computational performance. More recently, [8] developed a modern PB method for solving either the convex or strongly convex version of (1) which: requires no knowledge of the Lipschitz parameters (M, L) and the strong convex parameter μ of ϕ; and allows the stepsize to change only at the beginning of each cycle. A potential drawback of the method of [8] is that it can restart a cycle with its current initial prox stepsize λ divided by two if λ is found to be large, i.e., the method can backtrack. In contrast, by allowing the prox stepsizes to vary within a cycle, Ad-GPB never has to restart a cycle.

In theory, classical PB methods perform on average $\mathcal{O}(\varepsilon^{-2})$ consecutive null iterations while modern PB methods perform only $\mathcal{O}(\varepsilon^{-1})$ consecutive null iterations in the worst case. The explanation for this phenomenon is due to the more relaxed δ-criterion used by modern PB methods to end a cycle. Our Ad-GPB method pursues the idea of further relaxing the cycle termination criterion to reduce its overall number of iterations, and hence improve its computational performance while retaining all the theoretical guarantees of the modern PB methods of [8, 18, 19]. More specifically, under the simplifying assumption that ϕ_* is known, an Ad-GPB cycle stops when, for some universal constant $\beta \in (0, 1]$, the inequality $\phi(y_j) - m_j \leq \delta + \beta[\phi(y_j) - \phi_*]$ is satisfied. The addition of the (usually large) term $\beta[\phi(y_j) - \phi_*]$ makes this inequality easier to satisfy, thereby resulting in Ad-GPB performing shorter cycles. Even though the previous observation assumes that ϕ_* is known, Ad-GPB removes this assumption, at the expense of assuming that the domain of h is bounded, by replacing ϕ_* in the above inequality with a suitable lower bound on ϕ_*.

Organization of the paper. Subsection 1.1 presents basic definitions and notation used throughout the paper. Section 2 contains two subsections. Subsection 2.1 formally describes problem (1) and the assumptions made on it. Subsection 2.2 presents a generic bundle update scheme, the Ad-GPB framework, and states the main iteration-complexity result for Ad-GPB. Section 3 provides a bound on the number of iterations within a cycle of Ad-GPB. Section 4 contains two subsections. The first (resp, second) one establishes bounds on the number of cycles and the total number of iterations performed by Ad-GPB under the assumption that ϕ_* is known (resp., unknown). Section 5 presents the numerical results comparing Ad-GPB with the two-cut bundle update scheme against other mordern PB methods.

1.1 Basic definitions and notation

The sets of real numbers and positive real numbers are denoted by $\mathbb{R}$ and $\mathbb{R}_{++}$, respectively. Let $\mathbb{R}^n$ denote the standard n-dimensional Euclidean space equipped with inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, respectively. Let $\log(\cdot)$ denote the natural logarithm and $\log^+(\cdot)$ denote $\max\{\log(\cdot), 0\}$. Let $\mathcal{O}$ denote the standard big-O notation.

For a given function $\varphi : \mathbb{R}^n \to (-\infty, +\infty]$, let $\operatorname{dom} \varphi := \{x \in \mathbb{R}^n : \varphi(x) < \infty\}$ denote the effective domain of φ and φ is proper if $\operatorname{dom} \varphi \neq \emptyset$. A proper function $\varphi : \mathbb{R}^n \to (-\infty, +\infty]$ is convex if

$$\varphi(\alpha x + (1 - \alpha)y) \leq \alpha\varphi(x) + (1 - \alpha)\varphi(y)$$

for every $x, y \in \operatorname{dom}\varphi$ and $\alpha \in [0,1]$. Denote the set of all proper lower semicontinuous convex functions by $\overline{\operatorname{Conv}}(\mathbb{R}^n)$.

The subdifferential of φ at $x \in \operatorname{dom}\varphi$ is denoted by

$$\partial\varphi(x) := \{s \in \mathbb{R}^n : \varphi(y) \geq \varphi(x) + \langle s, y - x\rangle, \forall y \in \mathbb{R}^n\}. \tag{3}$$

The set of proper closed convex functions Γ such that $\Gamma \leq \phi$ is denoted by $\mathcal{B}(\phi)$ and any such Γ is called a bundle for ϕ.

The sign function $\operatorname{sign} : \mathrm{R^n} \to \mathrm{R^n}$ is defined as

$$\operatorname{sign}(\mathrm{x})_\mathrm{i} = \begin{cases} -1, & \text{if } x_i < 0, \\ 0, & \text{if } x_i = 0, \\ 1, & \text{if } x_i > 0. \end{cases}$$

2 Main problem and algorithm

This section contains two subsections. The first one describes the main problem and corresponding assumptions. The second one presents the motivation and the description of Ad-GPB, as well as its main complexity result.

2.1 Main problem

The problem of interest in this paper is (1) which is assumed to satisfy the following conditions for some constants $M \geq 0$ and $L \geq 0$:

(A1) $h \in \overline{\operatorname{Conv}}(\mathbb{R}^n)$ and there exists $D \geq 0$ such that

$$\sup_{x,y \in \operatorname{dom} h} \|y - x\| \leq D; \tag{4}$$

(A2) $f \in \overline{\operatorname{Conv}}(\mathbb{R}^n)$ is such that $\operatorname{dom} h \subset \operatorname{dom} f$, and a subgradient oracle, i.e., a function $f' : \operatorname{dom} h \to \mathbb{R}^n$ satisfying $f'(x) \in \partial f(x)$ for every $x \in \operatorname{dom} h$, is available;

(A3) for every $x, y \in \operatorname{dom} h$,

$$\|f'(x) - f'(y)\| \leq 2M + L\|x - y\|.$$

In addition to the above assumptions, it is also assumed that h is simple in the sense that, for any $\lambda > 0$ and affine function $\mathcal{A}$, the following two optimization problems

$$\min_u \mathcal{A}(u) + h(u), \qquad \min_u \mathcal{A}(u) + h(u) + \frac{1}{2\lambda}\|u\|^2 \tag{5}$$

are easy to solve.

We now make three remarks about assumptions (A1)-(A3). First, it can be shown that (A1) implies that both problems in (5) have optimal solutions. Second, it can also be shown that (A1) implies that the set of optimal solutions X^* of problem (1) is nonempty. Third, letting $\tilde{\ell}_f(\cdot; x)$ denotes the linearization of f at x, i.e.,

$$\tilde{\ell}_f(\cdot; x) := f(x) + \langle f'(x), \cdot - x\rangle \quad \forall x \in \operatorname{dom} h, \tag{6}$$

then it is well-known that (A3) implies that for every $x, y \in \operatorname{dom} h$,

$$f(x) - \tilde{\ell}_f(x; y) \leq 2M\|x - y\| + \frac{L}{2}\|x - y\|^2. \tag{7}$$

Finally, define the composite linearization of the objective ϕ of (1) at x as

$$\ell_\phi(\cdot; x) := \tilde{\ell}_f(\cdot; x) + h(\cdot) \quad \forall x \in \operatorname{dom} h. \tag{8}$$

2.2 Algorithm

As mentioned in the Introduction, PB uses a bundle (convex) function underneath $\phi(\cdot)$ to construct subproblem (2) at a given iteration, and then updates Γ_j to obtain the bundle function Γ_{j+1} for the next iteration. This subsection describes ways of updating the bundle. Instead of focusing on a specific bundle update scheme, this subsection describes a generic bundle update framework (BUF) which is a restricted version of the one introduced in Subsection 3.1 of [19]. It also discusses two concrete bundle update schemes lying within the framework.

We start by describing the generic BUF.

BUF

Input: $\lambda \in \mathbb{R}_{++}$ and $(x^c, x, \Gamma) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathcal{B}(\phi)$ such that

$$x = \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \Gamma(u) + \frac{1}{2\lambda}\|u - x^c\|^2 \right\}, \tag{9}$$

- find bundle $\Gamma^+ \in \mathcal{B}(\phi)$ satisfying

$$\Gamma^+(\cdot) \geq \max\{\bar{\Gamma}(\cdot), \ell_\phi(\cdot; x)\}, \tag{10}$$

for some $\bar{\Gamma}(\cdot) \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n)$ such that

$$\bar{\Gamma}(x) = \Gamma(x), \quad x = \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \bar{\Gamma}(u) + \frac{1}{2\lambda}\|u - x^c\|^2 \right\}. \tag{11}$$

Output: Γ^+.

Now we make some remarks about BUF. First, observe that if $\Gamma \in \mathcal{B}(\phi)$ and $\Gamma \geq \ell_\phi(\cdot; \bar{x})$ for some $\bar{x} \in \mathbb{R}^n$, then $\operatorname{dom}\Gamma = \operatorname{dom} h$. Hence, it follows from (10) and the definition of $\mathcal{B}(\phi)$ that the output Γ^+ of BUF satisfies

$$\Gamma^+ \leq \phi, \quad \Gamma^+ \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n), \quad \operatorname{dom}\Gamma^+ = \operatorname{dom} h.$$

Second, the bundle update framework of [19] replaces (10) by the weaker inequality $\Gamma^+(\cdot) \geq \tau\bar{\Gamma}(\cdot) + (1 - \tau)\ell_\phi(\cdot; x)$ for some $\tau \in (0, 1)$ and, as a result, contains the one-cut bundle update scheme described in Subsection 3.1 of [19]. Even though BUF does not include the one-cut bundle update scheme, it contains the two other bundle update schemes discussed in [19] (see Subsection 3.1), which for convenience are briefly described below.

- **2-cut:** For this scheme, it is assumed that Γ has the form

$$\Gamma = \max\{A_f, \tilde{\ell}_f(\cdot; x^-)\} + h \tag{12}$$

where $h \in \operatorname{Conv}(\mathbb{R}^n)$ and A_f is an affine function satisfying $A_f \leq f$. In view of (9), it can be shown that there exists $\theta \in [0, 1]$ such that

$$\frac{1}{\lambda}(x - x^c) + \partial h(x) + \theta\nabla A_f + (1 - \theta)f'(x^-) \ni 0, \tag{13}$$

$$\theta A_f(x) + (1 - \theta)\ell_f(x; x^-) = \max\{A_f(x), \tilde{\ell}_f(x; x^-)\}. \tag{14}$$

The scheme then sets

$$A_f^+(\cdot) := \theta A_f(\cdot) + (1 - \theta)\tilde{\ell}_f(\cdot; x^-) \tag{15}$$

and outputs the function Γ^+ defined as

$$\Gamma^+(\cdot) := \max\{A_f^+(\cdot), \tilde{\ell}_f(\cdot; x)\} + h(\cdot). \tag{16}$$

- **multiple-cut (M-cut):** Suppose $\Gamma = \Gamma(\cdot; C)$ where $C \subset \mathbb{R}^n$ is a finite set (i.e., the current bundle set) and $\Gamma(\cdot; C)$ is defined as

$$\Gamma(\cdot; C) := \max\{\tilde{\ell}_f(\cdot; c) : c \in C\} + h(\cdot). \tag{17}$$

This scheme chooses the next bundle set C^+ so that

$$C(x) \cup \{x\} \subset C^+ \subset C \cup \{x\} \tag{18}$$

where

$$C(x) := \{c \in C : \tilde{\ell}_f(x; c) + h(x) = \Gamma(x)\}, \tag{19}$$

and then output $\Gamma^+ = \Gamma(\cdot; C^+)$.

The following facts, whose proofs can be found in Appendix D of [19], imply that 2-cut and M-cut schemes are special implementations of BUF:

(a) If Γ^+ is obtained according to 2-cut, then $(\Gamma^+, \bar{\Gamma})$ where $\bar{\Gamma} = A_f^+ + h$ satisfies (10) and (11);

(b) If Γ^+ is obtained according to M-cut, then $(\Gamma^+, \bar{\Gamma})$ where $\bar{\Gamma} = \Gamma(\cdot; C(x))$ satisfies (10) and (11).

We next give an outline of Ad-GPB. Ad-GPB is an inexact proximal point method which, given the $(k-1)$-th prox center $\hat{x}_{k-1} \in \mathbb{R}^n$, finds a pair $(\hat{x}_k, \hat{\lambda}_k)$ of prox stepsize $\hat{\lambda}_k > 0$ and k-th prox-center $\hat{x}_k$ satisfying a suitable error criterion for being an approximate solution of the prox subproblem

$$\hat{x}_k \approx \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \phi(u) + \frac{1}{2\hat{\lambda}_k} \|u - \hat{x}_{k-1}\|^2 \right\}. \tag{20}$$

More specifically, Ad-GPB solves a sequence of prox bundle subproblems

$$x_j = \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \Gamma_j(u) + \frac{1}{2\lambda_j} \|u - \hat{x}_{k-1}\|^2 \right\}, \tag{21}$$

where Γ_j is a bundle approximation of ϕ and $\lambda_j \le \lambda_{j-1}$ is an adaptively chosen prox stepsize, until the pair $(\hat{x}_k, \hat{\lambda}_k) = (x_j, \lambda_j)$ satisfy the approximate error criterion for (20). In contrast to the GPB method of [19], which can also be viewed in the setting outlined above, Ad-GPB: i) (adaptively) changes λ_j while computing the next prox center $\hat{x}_k$; and, ii) Ad-GPB stops the search for the next prox center $\hat{x}_k$ using a termination criterion based not only on the user-provided tolerance (the quantity ε in the description below) as GPB also does, but also on a suitable primal-dual gap for (20), a feature that considerably speeds up the computation of $\hat{x}_k$ for many subproblems (20).

We now formally describe Ad-GPB. Its description uses the definition of the set of bundles $\mathcal{B}(\phi)$ for the function ϕ given in Subsection 1.1.

Ad-GPB

---

0. Let $\hat{x}_0 \in \operatorname{dom} h$, $\lambda_1 > 0$, $0 \le \beta_0 \le 1/2$, $\tau \in (0,1)$, and $\varepsilon > 0$ be given; find $\Gamma_1 \in \mathcal{B}(\phi)$ such that $\Gamma_1 \ge \ell_\phi(\cdot; \hat{x}_0)$ and set $\hat{\ell}_0 = \min_u \Gamma_1(u)$, $y_0 = \hat{x}_0$, $j_0 = 0$, $j = k = 1$, and $\hat{n}_0 = \hat{\ell}_0$;

1. compute x_j as in (21) and

$$m_j := \Gamma_j(x_j) + \frac{1}{2\lambda_j} \|x_j - \hat{x}_{k-1}\|^2 \tag{22}$$

$$y_j := \operatorname{argmin} \{\phi(x) : x \in \{y_{j-1}, x_j\}\} \tag{23}$$

$$t_j := \phi(y_j) - m_j; \tag{24}$$

2. **if** $t_j \le \beta_{k-1}[\phi(y_j) - \hat{n}_{k-1}] + \varepsilon/4$ is violated **then** perform a **null update**, i.e.:
if either $j = j_{k-1} + 1$ or

$$t_j - \tau t_{j-1} \le (1-\tau)\left\{\frac{\beta_{k-1}[\phi(y_j) - \hat{n}_{k-1}]}{2} + \frac{\varepsilon}{8}\right\}, \tag{25}$$

then set $\lambda_{j+1} = \lambda_j$; else, set $\lambda_{j+1} = \lambda_j/2$;
set $\Gamma_{j+1} = \mathrm{BUF}(\hat{x}_{k-1}, x_j, \Gamma_j, \lambda_j)$;
else perform a **serious update**, i.e.:
set $(\hat{\lambda}_k, \hat{x}_k, \hat{y}_k, \hat{\Gamma}_k, \hat{m}_k, \hat{t}_k) = (\lambda_j, x_j, y_j, \Gamma_j, m_j, t_j)$;
compute

$$\hat{\Gamma}_k^a(u) := \frac{\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l \hat{\Gamma}_l(u)}{\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l}, \quad \hat{\ell}_k := \max\left\{\hat{\ell}_{k-1}, \inf_u \hat{\Gamma}_k^a(u)\right\}, \tag{26}$$

and choose $\hat{n}_k \in [\hat{\ell}_k, \phi_*]$;
if $\phi(\hat{y}_k) - \hat{n}_k \le \varepsilon$, output $(\hat{x}_k, \hat{y}_k)$, and **stop**; **else** compute

$$\hat{\phi}_k^a := \frac{\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l \phi(\hat{y}_l)}{\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l}, \tag{27}$$

$$\hat{g}_k := \frac{1}{\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l} \sum_{l=\lceil k/2\rceil}^{k} \beta_{l-1}\hat{\lambda}_l[\phi(\hat{y}_l) - \hat{n}_{l-1}]; \tag{28}$$

if $\hat{g}_k \le (\hat{\phi}_k^a - \hat{n}_k)/2$, then set $\beta_k = \beta_{k-1}$; **else** set $\beta_k = \beta_{k-1}/2$;
set $\lambda_{j+1} = \lambda_j$ and find $\Gamma_{j+1} \in \mathcal{B}(\phi)$ such that $\Gamma_{j+1} \ge \ell_\phi(\cdot; x_j)$;
set $j_k = j$ and $k \leftarrow k+1$;
end if

3. set $j \leftarrow j+1$ and go to step 1.

We now introduce some terminology related to Ad-GPB. Ad-GPB performs two types of iterations, namely, null and serious, corresponding to the kinds of updates performed at the end. The index j counts the iterations (including null and serious). Let $j_1 \le j_2 \le \ldots$ denote the sequence of all serious iterations (i.e., the ones ending with a serious update) and, for every $k \ge 1$, define $i_k = j_{k-1} + 1$ and the k-th cycle $\mathcal{C}_k$ as

$$\mathcal{C}_k := \{i_k, \ldots, j_k\}. \tag{29}$$

Observe that for every $k \ge 1$, we have $\hat{\lambda}_k = \lambda_{j_k}$ where $\hat{\lambda}_k$ is computed in the serious update part of step 2 of Ad-GPB. (Hence, index k counts the cycles generated by Ad-GPB.) An iteration j is called good (resp., bad) if $\lambda_{j+1} = \lambda_j$ (resp., $\lambda_{j+1} = \lambda_j/2$). Note that the logic of Ad-GPB implies that i_k and j_k are good iterations and that (25) is violated whenever j is a bad iteration.

We next make some remarks about the quantities related to different Γ-functions that appear in Ad-GPB and the associated quantities $\hat{\ell}_k$ and $\hat{n}_k$. First, the observation immediately following BUF implies that

$$\Gamma_j \le \phi, \quad \Gamma_j \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n), \quad \mathrm{dom}\,\Gamma_j = \mathrm{dom}\,h \quad \forall j \ge 1, \tag{30}$$

which together with the fact that $\hat{\Gamma}_k$ is the last Γ_j generated within a cycle imply that $\hat{\Gamma}_k \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n)$ and $\hat{\Gamma}_k \le \phi$. Moreover, the first identity in (26) and the latter conclusion then imply that $\hat{\Gamma}_k^a \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n)$ and $\hat{\Gamma}_k^a \le \phi$, and hence that $\inf_u \hat{\Gamma}_k^a(u) \le \inf_u \phi(u) = \phi_*$. Second, the facts that $\mathrm{dom}\,\hat{\Gamma}_k^a = \mathrm{dom}\,h$ is bounded (see assumption A1) and $\hat{\Gamma}_k^a$ is a closed convex function imply that $\inf_u \hat{\Gamma}_k^a(u) > -\infty$. Moreover, the problem $\inf_u \hat{\Gamma}_k^a(u)$ has the same format as the first one that appears in (5), and hence is easily solvable by assumption. Its optimal value, which is a lower bound on ϕ_* as already observed above, is used to update the lower bound

$\hat{\ell}_{k-1}$ for ϕ_* to a possibly sharper one, namely, $\hat{\ell}_k \geq \hat{\ell}_{k-1}$. Thus, the choice of $\hat{n}_k$ in the line following (26) makes sense. For the sake of future reference, we note that

$$\phi_* \geq \hat{n}_k \geq \hat{\ell}_k \geq \inf_u \hat{\Gamma}_k^a(u). \tag{31}$$

Third, obvious ways of choosing $\hat{n}_k$ in the interval $[\hat{\ell}_k, \phi_*]$ are: i) $\hat{n}_k = \phi_*$; and ii) $\hat{n}_k = \hat{\ell}_k$. While choice i) requires knowledge of ϕ_*, choice ii) does not and can be easily implemented in view of the previous remark. Moreover, if ϕ_* is known and $\hat{n}_k$ is chosen as in i), then there is no need to compute $\hat{l}_k$, and hence the min term in (26), at the end of every cycle.

We now make some other relevant remarks about Ad-GPB. First, it follows from (30) and the definition of x_j in (21) that $x_j \in \operatorname{dom} h$ for every $j \geq 1$. Second, an induction argument using (23) and the fact that $y_0 = \hat{x}_0 \in \operatorname{dom} h$ imply that $y_j \in \{\hat{x}_0, x_1, \ldots, x_j\} \subset \operatorname{dom} h$ and

$$y_j \in \operatorname{Argmin} \{\phi(x) : x \in \{\hat{x}_0, x_1, \ldots, x_j\}\} \tag{32}$$

(hence, $\phi(y_{j+1}) \leq \phi(y_j)$) for every $j \geq 1$. Third, the cycle-stopping criterion, i.e., the inequality in the first line of step 2, is a relaxation of the one used by GPB method of [19], in the sense its right-hand side has the extra term $\beta_{k-1}[\phi(y_j) - \hat{n}_{k-1}]$ involving the relaxation factor β_{k-1}. The addition of this term allows earlier cycles to terminate in less number of inner iterations, and hence speeds up the overall performance of the method. The quantities in (27) and (28) are used to update β_{k-1} at the end of the k-th cycle (see 'if' statement after (28)). Fourth, the condition imposed on Γ_{j+1} at the end of a serious iteration (see the second line below (28)) does not completely specify it. An obvious way (cold start) of choosing this Γ_{j+1} is to set it to be $\ell_\phi(\cdot; x_j)$; another way (warm start) is to choose it using the update rule of an null iteration, i.e., $\Gamma_{j+1} = \mathrm{BUF}(\hat{x}_{k-1}, x_j, \Gamma_j, \lambda_j)$ since (10) implies the required condition on Γ_{j+1}.

We now comment on the inexactness of $\hat{y}_k$ as a solution of prox subproblem (20) and as a solution of (1) upon termination of Ad-GPB. The fact that $\hat{\Gamma}_k \leq \phi$ and the fact that $\hat{t}_k = t_{j_k}$ imply that the primal gap of (20) at $\hat{y}_k$ is upper bounded by $\hat{t}_k + \|\hat{y}_k - \hat{x}_{k-1}\|^2/(2\hat{\lambda}_k)$. Hence, if the inequality for stopping the cycle in step 2 holds, then we conclude that $\hat{y}_k$ is an ε_k-solution of (20), where

$$\varepsilon_k := \frac{\varepsilon}{4} + \beta_{k-1}[\phi(\hat{y}_k) - \hat{n}_{k-1}] + \frac{\|\hat{y}_k - \hat{x}_{k-1}\|^2}{2\hat{\lambda}_k}.$$

Finally, if the test inequality before (27) in step 2 holds, then the second component $\hat{y}_k$ of the pair output by Ad-GPB satisfies $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$ due to the fact that $\hat{n}_k \leq \phi_*$.

Lastly, Ad-GPB never restarts a cycle, i.e., attempts to inexactly solve two or more subproblems (20) with the same prox center $\hat{x}_{k-1}$. Instead, Ad-GPB has a key rule for updating the inner stepsize λ_j which always allows it to inexactly solve subproblem (20) with $\hat{\lambda}_k$ set to be the last λ_j generated within the k-th cycle (see the second line of the serious update part of Ad-GPB).

We now state the main complexity result for Ad-GPB whose proof is postponed to the end of Section 4.

Theorem 2.1 *Define*

$$\bar{t} := 2MD + \frac{L}{2}D^2, \tag{33}$$

$$\bar{K}(\varepsilon) := 2\left\lceil \frac{2D^2\bar{Q}}{\varepsilon}\left(\frac{M^2}{\varepsilon} + \frac{L}{16} + \frac{1}{\lambda_1\bar{Q}}\right) + \log^+\left\{\frac{\beta_0(\phi(\hat{x}_0) - \hat{n}_0)}{\varepsilon}\right\} + 1\right\rceil \tag{34}$$

where

$$\bar{Q} = \frac{128(1-\tau)}{\tau}. \tag{35}$$

Then, Ad-GPB finds a pair $(\hat{y}_k, \hat{n}_k) \in \operatorname{dom}\phi \times \mathbb{R}$ *satisfying* $\phi(\hat{y}_k) - \phi_* \leq \phi(\hat{y}_k) - \hat{n}_k \leq \varepsilon$ *in at most* $4\bar{K}(\varepsilon)$ *cycles and*

$$4\bar{K}(\varepsilon)\left(\frac{1+\tau}{1-\tau}\log^+\left[8\bar{t}\varepsilon^{-1}\right] + 2\right) + \log_2^+\left(\bar{Q}\lambda_1\left(\frac{M^2}{\varepsilon} + \frac{L}{16}\right)\right) \tag{36}$$

iterations.

We now make some remarks about Theorem 2.1. First, in terms of τ and ε only, it follows from Theorem 2.1 that the iteration complexity of Ad-GPB to find a ε-solution of (1) is $\tilde{\mathcal{O}}\left(\varepsilon^{-2}\tau^{-1} + (1-\tau)^{-1})\right)$. Hence, when $\tau \in (0,1)$ satisfies $\tau^{-1} = \mathcal{O}(1)$ and $\tau = 1 - \Omega(\varepsilon^2)$, the total iteration complexity of Ad-GPB is $\tilde{\mathcal{O}}(\varepsilon^{-2})$. Moreover, under the assumption that $\tau \in (0,1)$ satisfies $\tau^{-1} = \mathcal{O}(1)$, Ad-GPB performs

- $\mathcal{O}(\varepsilon^{-2})$ cycles whenever $\tau = 1 - \Theta(1)$;
- more generally, $\mathcal{O}(\varepsilon^{-\alpha})$ cycles whenever $\tau = 1 - \Theta(\varepsilon^{2-\alpha})$ for some $\alpha \in [0,2]$.

3 Bounding cycle lengths

The main goal of this section is to derive a bound (Proposition 3.5 below) on the number of iterations within a cycle.

Recall from (29) that i_k (resp., j_k) denotes the first (resp., last) iteration index of the k-th cycle of Ad-GPB. The first result describes some basic facts about the iterations within any given cycle.

Lemma 3.1 *For every $j \in \mathcal{C}_k \setminus \{i_k\}$, the following statements hold:*

a) there exists function $\bar{\Gamma}_{j-1}(\cdot)$ such that

$$\max\left\{\bar{\Gamma}_{j-1}(\cdot), \ell_f(\cdot; x_{j-1}) + h(\cdot)\right\} \le \Gamma_j(\cdot) \le \phi(\cdot), \tag{37}$$

$$\bar{\Gamma}_{j-1} \in \mathrm{Conv}(\mathbb{R}^n), \quad \bar{\Gamma}_{j-1}(x_{j-1}) = \Gamma_{j-1}(x_{j-1}), \tag{38}$$

$$x_{j-1} = \operatorname*{argmin}_{u\in\mathbb{R}^n}\left\{\bar{\Gamma}_{j-1}(u) + \frac{1}{2\lambda_{j-1}}\|u - \hat{x}_{k-1}\|^2\right\}; \tag{39}$$

b) for every $u \in \mathbb{R}^n$, we have

$$\bar{\Gamma}_{j-1}(u) + \frac{1}{2\lambda_{j-1}}\|u - \hat{x}_{k-1}\|^2 \ge m_{j-1} + \frac{1}{2\lambda_{j-1}}\|u - x_{j-1}\|^2. \tag{40}$$

Proof: a) This statement immediately follows from (10), (11), and the facts that Γ_j is the output of the BUF blackbox with input λ_{j-1} and $(x^c, x, \Gamma) = (x^c_{j-1}, x_{j-1}, \Gamma_{j-1})$ and $x^c_{j-1} = \hat{x}_{k-1}$.

b) Using (39) and the fact that $f = \bar{\Gamma}_{j-1} + \|\cdot - \hat{x}_{k-1}\|^2/(2\lambda_{j-1})$ is λ_{j-1}^{-1} strongly convex, we have for every $u \in \mathrm{dom}\, h$,

$$\bar{\Gamma}_{j-1}(u) + \frac{1}{2\lambda_{j-1}}\|u - \hat{x}_{k-1}\|^2 \ge \bar{\Gamma}_{j-1}(x_{j-1}) + \frac{1}{2\lambda_{j-1}}\|x_{j-1} - \hat{x}_{k-1}\|^2 + \frac{1}{2\lambda_{j-1}}\|u - x_{j-1}\|^2.$$

The statement follows from the above inequality and the second identity in (38). ∎

The next result presents some basic recursive inequalities for $\{t_j\}$.

Lemma 3.2 *For every $j \in \mathcal{C}_k \setminus \{i_k\}$, the following statements hold:*

a) for every $\tau' \in [0,1]$, there holds

$$t_j - \tau' t_{j-1} \le 2M(1-\tau')\|x_j - x_{j-1}\| - \left(\frac{\tau'}{2\lambda_{j-1}} - \frac{(1-\tau')L}{2}\right)\|x_j - x_{j-1}\|^2 - \frac{1-\tau'}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2;$$

b) if $\lambda_{j-1} \le \tau/(2(1-\tau)L)$, then we have

$$t_j - \tau t_{j-1} \le \frac{4M^2(1-\tau)^2\lambda_{j-1}}{\tau}. \tag{41}$$

Proof: a) Inequality (37) implies that for every $\tau' \in [0,1]$, we have

$$\Gamma_j(x_j) \geq \max\left\{\bar{\Gamma}_{j-1}(x_j), \ell_\phi(x_j; x_{j-1})\right\} \geq (1-\tau')\ell_\phi(x_j; x_{j-1}) + \tau'\bar{\Gamma}_{j-1}(x_j).$$

The definition of m_j in (22), the above inequality, and (40) with $u = x_j$, imply that

$$\begin{aligned}
m_j &\geq (1-\tau')\ell_\phi(x_j; x_{j-1}) + \tau'\bar{\Gamma}_{j-1}(x_j) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2 \\
&= (1-\tau')\left[\ell_\phi(x_j; x_{j-1}) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] + \tau'\left[\bar{\Gamma}_{j-1}(x_j) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] \\
&\overset{\lambda_j \leq \lambda_{j-1}}{\geq} (1-\tau')\left[\ell_\phi(x_j; x_{j-1}) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] + \tau'\left[\bar{\Gamma}_{j-1}(x_j) + \frac{1}{2\lambda_{j-1}}\|x_j - \hat{x}_{k-1}\|^2\right] \\
&\overset{(40)}{\geq} (1-\tau')\left[\ell_\phi(x_j; x_{j-1}) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] + \tau'\left[m_{j-1} + \frac{1}{2\lambda_{j-1}}\|x_j - x_{j-1}\|^2\right].
\end{aligned}$$

Using this inequality and the definition of t_j in (24), we have

$$\begin{aligned}
t_j - \tau' t_{j-1} &= [\phi(y_j) - m_j] - \tau'[\phi(y_{j-1}) - m_{j-1}] \\
&= [\phi(y_j) - \tau'\phi(y_{j-1})] - [m_j - \tau' m_{j-1}] \\
&\leq [\phi(y_j) - \tau'\phi(y_{j-1})] - (1-\tau')\left[\ell_\phi(x_j; x_{j-1}) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] - \frac{\tau'}{2\lambda_{j-1}}\|x_j - x_{j-1}\|^2 \\
&= [\phi(y_j) - \tau'\phi(y_{j-1}) - (1-\tau')\phi(x_j)] \\
&\quad + (1-\tau')\left[\phi(x_j) - \ell_\phi(x_j; x_{j-1})\right] - \frac{1-\tau'}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2 - \frac{\tau'}{2\lambda_{j-1}}\|x_j - x_{j-1}\|^2 \\
&\leq (1-\tau')\left[\phi(x_j) - \ell_\phi(x_j; x_{j-1})\right] - \frac{1-\tau'}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2 - \frac{\tau'}{2\lambda_{j-1}}\|x_j - x_{j-1}\|^2,
\end{aligned}$$

where the last inequality is due to the definition of y_j in (23). The conclusion of the statement now follows from the above inequality and relation (7) with $(y, x) = (x_{j-1}, x_j)$.

b) Using the assumption of this statement and statement a) with $\tau' = \tau$, we easily see that

$$t_j - \tau t_{j-1} \leq 2M(1-\tau)\|x_j - x_{j-1}\| - \frac{\tau}{4\lambda_{j-1}}\|x_j - x_{j-1}\|^2.$$

The statement now follows from the above inequality and the inequality $2ab - b^2 \leq a^2$ with

$$a = \frac{2M(1-\tau)\sqrt{\lambda_{j-1}}}{\sqrt{\tau}}, \quad b = \frac{\sqrt{\tau}\|x_j - x_{j-1}\|}{2\sqrt{\lambda_{j-1}}}.$$

■

The next result describes some properties about the stepsizes λ_j within any given cycle. It uses the fact that if j is a bad iteration of Ad-GPB, then (25) is violated (see step 2 of Ad-GPB and the first paragraph following Ad-GPB).

Lemma 3.3 *Define*

$$\underline{\lambda} := \min\left\{\frac{\tau\varepsilon}{128(1-\tau)M^2}, \frac{\tau}{8(1-\tau)L}\right\}; \tag{42}$$

where τ is an input to Ad-GPB, and M and L are as in Assumption 3. Then, the following statements hold:

a) for every index $j \in \mathcal{C}_k$, we have

$$\lambda_j \geq \min\left\{\underline{\lambda}, \lambda_{i_k}\right\};$$

b) the number of bad iterations in $\mathcal{C}_k$ is bounded by $\log_2^+(\lambda_{i_k}/\underline{\lambda})$.

Proof: a) Assume for contradiction that there exists $j \in \mathcal{C}_k$ such that

$$\lambda_j < \min\{\underline{\lambda}, \lambda_{i_k}\}, \tag{43}$$

and that j is the smallest index in $\mathcal{C}_k$ satisfying the above inequality. We claim that this assumption implies that

$$\frac{\lambda_{j-2}}{4} \leq \frac{\lambda_{j-1}}{2} = \lambda_j. \tag{44}$$

Before showing the claim, we argue that (44) implies the conclusion of the lemma. Indeed, noting that (42) and (44) implies that $\lambda_{j-2} \leq 4\underline{\lambda} \leq \tau/(2(1-\tau)L)$, it follows from (41) with $j = j-1$ and the definition of $\underline{\lambda}$ in (42) that

$$\begin{aligned} t_{j-1} - \tau t_{j-2} &\overset{(41)}{\leq} \frac{4(1-\tau)^2\lambda_{j-2}M^2}{\tau} \leq \frac{16(1-\tau)^2\lambda_j M^2}{\tau} \leq \frac{16(1-\tau)^2\underline{\lambda}M^2}{\tau} \\ &\overset{(42)}{\leq} (1-\tau)\frac{\varepsilon}{8} \leq (1-\tau)\left(\frac{\beta_{k-1}(\phi(y_{j-1}) - \hat{n}_{k-1})}{2} + \frac{\varepsilon}{8}\right), \end{aligned}$$

where the last inequality is due to the fact that $\phi(y_{j-1}) - \hat{n}_{k-1} \geq 0$. This conclusion then implies that (25) holds for iteration $j-1$, and hence that $\lambda_j = \lambda_{j-1}$ due to the logic of step 2 of Ad-GPB. Since this contradicts (44), statement (a) follows.

We will now show the above claim, i.e., that the definition of j implies (44). Indeed, since the logic of step 2 implies that $\lambda_{i_k+1} = \lambda_{i_k}$ and j is the smallest index in $\mathcal{C}_k$ satisfying (43), we conclude that $j \geq i_k + 2$ and $\lambda_j \neq \lambda_{j-1}$. Using these conclusions and the fact that the logic of step 2 of Ad-GPB implies that either $\lambda_i = \lambda_{i-1}$ or $\lambda_i = \lambda_{i-1}/2$ for every $i \in \mathcal{C}_k \setminus \{i_k\}$, we then conclude that both the inequality and the identity in (44) hold.

b) Since $\lambda_{j+1} = \lambda_j$ (resp., $\lambda_{j+1} = \lambda_j/2$) if j is a good (resp., bad) iteration, we easily see that $\lambda_{i_k}/\hat{\lambda}_k = 2^{s_k}$. This observation together with (a) then implies that statement (b) holds. ∎

It follows from Lemma 3.3(b) that the number of bad iterations within the k-th cycle $\mathcal{C}_k$ is finite. Proposition 3.5 below provides a bound on $|\mathcal{C}_k|$, and hence shows that every cycle $\mathcal{C}_k$ terminates. Before showing this result, we state a technical result which provides some key properties about the sequence $\{t_j\}$.

Lemma 3.4 *The following statements hold:*

a) if $j \in \mathcal{C}_k \setminus \{i_k\}$, then $t_j \leq t_{j-1}$.

b) if $j \in \mathcal{C}_k \setminus \{i_k\}$ is a good iteration that is not the last one in $\mathcal{C}_k$, then

$$t_j - \frac{\varepsilon}{8} \leq \frac{2\tau}{1+\tau}\left(t_{j-1} - \frac{\varepsilon}{8}\right);$$

c) if $j \in \mathcal{C}_k$ is not the last iteration of $\mathcal{C}_k$, then

$$t_j - \frac{\varepsilon}{8} \leq \left(\frac{2\tau}{1+\tau}\right)^{j-i_k-s_k}\left(t_{i_k} - \frac{\varepsilon}{8}\right) \tag{45}$$

where s_k denotes the number of bad iterations within cycle k.

Proof: a) The statement immediately follows from Lemma 3.2(a) with $\tau' = 1$.

b) Assume that $j \in \mathcal{C}_k \setminus \{i_k\}$ is a good iteration that is not the last one in $\mathcal{C}_k$. This together with the logic of the Ad-GPB imply that (25) is satisfied and the cycle-stopping criterion is violated at iteration j, i.e.,

$$t_j - \frac{\varepsilon}{4} > \beta_{k-1}(\phi(y_j) - \hat{n}_{k-1}). \tag{46}$$

These two observations then imply that

$$\begin{aligned} t_j - \frac{\varepsilon}{8} &\overset{(25)}{\leq} \tau t_{j-1} + (1-\tau)\left[\frac{\beta_{k-1}(\phi(y_j) - \hat{n}_{k-1})}{2} + \frac{\varepsilon}{8}\right] - \frac{\varepsilon}{8} \\ &= \tau\left(t_{j-1} - \frac{\varepsilon}{8}\right) + \frac{1-\tau}{2}\left[\beta_{k-1}(\phi(y_j) - \hat{n}_{k-1})\right] \\ &\overset{(46)}{\leq} \tau\left(t_{j-1} - \frac{\varepsilon}{8}\right) + \frac{1-\tau}{2}\left(t_j - \frac{\varepsilon}{4}\right) \leq \tau\left(t_{j-1} - \frac{\varepsilon}{8}\right) + \frac{1-\tau}{2}\left(t_j - \frac{\varepsilon}{8}\right), \end{aligned}$$

which can be easily seen to imply that statement b) holds.

c) If $j - i_k - s_k \leq 0$, then (45) obviously follows. Assume then that $j - i_k - s_k > 0$. The fact that there are at least $j - i_k - s_k$ good iterations in $\{i_k + 1, \ldots, j\}$, and statements a) and b), imply that

$$t_j - \frac{\varepsilon}{8} \leq \left(t_{i_k} - \frac{\varepsilon}{8}\right)\left(\frac{2\tau}{1+\tau}\right)^{j - i_k - s_k}. \tag{47}$$

and thus (45) follows. ∎

Proposition 3.5 *For every cycle index $k \geq 1$ generated by Ad-GPB, its size is bounded by $|\mathcal{C}_k| \leq s_k + \bar{N}_k(\varepsilon) + 1$ where s_k denotes the number of bad iterations within it and $\bar{N}_k(\cdot)$ is defined as*

$$\bar{N}_k(\varepsilon) := \left\lceil \frac{1+\tau}{1-\tau} \log^+\left[8 t_{i_k} \varepsilon^{-1}\right] \right\rceil. \tag{48}$$

Proof: If $t_{i_k} < \varepsilon/8$, then the cycle-stopping criterion is satisfied with $j = i_k$. This implies that $|\mathcal{C}_k| = 1$, and hence that the result trivially holds in this case. From now on, assume $t_{i_k} \geq \varepsilon/8$ and suppose for contradiction that $|\mathcal{C}_k| > s_k + \bar{N}_k(\varepsilon) + 1$. This implies that there exists a nonnegative integer $J \geq i_k$ such that $J + 1 \in \mathcal{C}_k$ and

$$J - i_k + 2 > s_k + \bar{N}_k(\varepsilon) + 1 \tag{49}$$

because the left-hand side of (49) is the cardinality of the index set $\{i_k, \ldots, J+1\}$. Since J is not the last iteration of $\mathcal{C}_k$, the cycle-stopping criterion in step 2 of Ad-GPB is violated with $j = J$, i.e.,

$$t_J > \beta_{k-1}[\phi(y_J) - \hat{n}_{k-1}] + \frac{\varepsilon}{4} \geq \frac{\varepsilon}{4}.$$

This observation together with Lemma 3.4(c) with $j = J$ then imply that

$$\frac{\varepsilon}{8} \leq t_J - \frac{\varepsilon}{8} \leq \left(\frac{2\tau}{1+\tau}\right)^{J - i_k - s_k}\left(t_{i_k} - \frac{\varepsilon}{8}\right) \leq \left(\frac{2\tau}{1+\tau}\right)^{J - i_k - s_k} t_{i_k},$$

which together with the definition of $\bar{N}_k(\varepsilon)$ in (48) can be easily seen to imply that

$$J - i_k - s_k \leq \bar{N}_k(\varepsilon) - 1.$$

Since this conclusion contradicts (49), the conclusion of the proposition follows. ∎

We now make some remarks about Proposition 3.5. First, the bound on the length of each cycle depends on the number of bad iterations within it. Second, to obtain the overall iteration complexity of Ad-GPB, it suffices to derive a bound on the number of cycles generated by Ad-GPB, which is the main goal of the subsequent section.

4 Bounding the number of cycles

This section establishes a bound on the number of cycles generated by Ad-GPB. It contains two subsections. The first one considers the (much simpler) case where ϕ_* is known and $\hat{n}_k$ in step 2 of Ad-GPB is set to ϕ_*

for every $k \geq 1$. The second one considers the general case where $\hat{n}_k$ is an arbitrary scalar satisfying the condition in step 2 of Ad-GPB.

We start by stating two technical results that are used in both subsections.

The first one describes basic facts about the sextuple $(\hat{\lambda}_k, \hat{x}_k, \hat{y}_k, \hat{\Gamma}_k, \hat{m}_k, \hat{t}_k)$ generated at the end of the k-th cycle.

Lemma 4.1 *For every cycle index k of Ad-GPB, the following statements hold:*

a) $\hat{\Gamma}_k \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n)$, $\hat{\Gamma}_k \leq \phi$, *and* $\mathrm{dom}\,\hat{\Gamma}_k = \mathrm{dom}\,h$;

b) we have

$$\hat{t}_k \leq \beta_{k-1}[\phi(\hat{y}_k) - \hat{n}_{k-1}] + \frac{\varepsilon}{4};$$

c) $\hat{x}_k, \hat{y}_k \in \mathrm{dom}\,h$, $\phi(\hat{y}_k) \leq \hat{\phi}_k^a$, *and* $\phi(\hat{y}_k) \leq \phi(\hat{y}_{k-1})$, *where by convention* $\hat{y}_0 = \hat{x}_0$;

d) $\hat{\ell}_k \geq \hat{\ell}_{k-1}$ *and* $\beta_k \leq \beta_{k-1}$;

e) for every given $u \in \mathrm{dom}\,h$, *we have*

$$\phi(\hat{y}_k) - \hat{\Gamma}_k(u) \leq \hat{t}_k + \frac{1}{2\hat{\lambda}_k}\left[\|u - \hat{x}_{k-1}\|^2 - \|u - \hat{x}_k\|^2\right]; \tag{50}$$

Proof: a) It follows from (30) and the fact that $\hat{\Gamma}_k$ is the last Γ_j generated within the k-th cycle.

b) It follows from the fact that the cycle-stopping criterion in the first line of step 2 of Ad-GPB is satisfied at iteration j_k and the definitions of the quantities $\hat{t}_k$ and $\hat{y}_k$ in step 2 of Ad-GPB.

c) It follows from the first two remarks in the paragraph containing (32), the definition of $\hat{\phi}_k^a$ in (27), and the fact that $\hat{x}_k$ (resp., $\hat{y}_k$) is the last x_j (resp., y_j) generated within the k-th cycle.

d) The first inequality in (d) follows from the definition $\hat{\ell}_k$ in (26). Moreover, the rule for updating β_k in step 2 of Ad-GPB implies that $\beta_k \leq \beta_{k-1}$.

e) Observe that (30) and the definitions of the quantities $\hat{x}_k$, $\hat{m}_k$, $\hat{\Gamma}_k$, and $\hat{\lambda}_k$ in step 2 of Ad-GPB, imply that $(\hat{x}_k, \hat{m}_k)$ is the pair of optimal solution and optimal value of

$$\min\left\{\hat{\Gamma}_k(u) + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2 : u \in \mathbb{R}^n\right\}. \tag{51}$$

The above observation, the fact that $\hat{\Gamma}_k(\cdot) + 1/(2\hat{\lambda}_k)\|\cdot - \hat{x}_{k-1}\|^2$ is $1/\hat{\lambda}_k$-strongly convex, together imply that, for the given $u \in \mathrm{dom}\,h$, we have

$$\hat{m}_k + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_k\|^2 \leq \hat{\Gamma}_k(u) + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2, \tag{52}$$

and hence that

$$\phi(\hat{y}_k) - \hat{\Gamma}_k(u) + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_k\|^2 \leq \phi(\hat{y}_k) - \hat{m}_k + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2 = \hat{t}_k + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2,$$

where the equality is due to the definition of $\hat{t}_k$ in step 2 of Ad-GPB. This shows that (50), and hence statement (e), holds. ∎

The next result provides a uniform upper (resp., lower) bound on the sequence $\{t_{i_k}\}$ (resp. $\hat{\lambda}_k$), and also a bound on the total number of bad iterations generated by Ad-GPB.

Lemma 4.2 *For every cycle index $k \geq 1$ generated by Ad-GPB, the following statements hold:*

a) $\hat{\lambda}_k \geq \min\{\underline{\lambda}, \lambda_1\}$ *where* $\underline{\lambda}$ *is as in* (42);

b) $\sum_{l=1}^{k} s_l \leq \log_2^+(\lambda_1/\underline{\lambda})$ *where s_l denotes the number of bad iterations within cycle l.*

c) we have $t_{i_k} \le \bar{t}$ where $\bar{t}$ is as in (33).

Proof: a) Using the facts that $\lambda_{i_k} = \hat{\lambda}_{k-1}$ (see step 2 of Ad-GPB) and Lemma 3.3(a) with $j = j_k$, we conclude that

$$\hat{\lambda}_k \ge \min\left\{\underline{\lambda}, \hat{\lambda}_{k-1}\right\}.$$

The statement then follows by using the above inequality recursively and the convention that $\hat{\lambda}_0 = \lambda_1$.

b) Since the last iteration of a cycle is not bad and λ_{j+1}/λ_j is equal to 1/2 (resp., equal to 1) if j is a bad iteration, we easily see that $\hat{\lambda}_l/\hat{\lambda}_{l-1} = (1/2)^{s_l}$, or equivalently, $\log_2 \hat{\lambda}_{l-1} - \log_2 \hat{\lambda}_l = s_l$, for every cycle l of Ad-GPB, under the convention that $\hat{\lambda}_0 := \lambda_1$. Statement (c) now follows by summing the above inequality from $l = 1$ to k and using statement a).

c) Using the facts that $\phi = f + h$ and $\Gamma_{i_k}(\cdot) \ge \tilde{\ell}_f(\cdot; \hat{x}_{k-1}) + h(\cdot)$ (see the serious update in step 2 of Ad-GPB), and the definition of t_j, m_j and y_j in (24), (22) and (23), respectively, we have

$$\begin{aligned} t_{i_k} &\overset{(24)}{=} \phi(y_{i_k}) - m_{i_k} \overset{(23)}{\le} \phi(x_{i_k}) - m_{i_k} \overset{(22)}{=} \phi(x_{i_k}) - \Gamma_{i_k}(x_{i_k}) - \frac{1}{2\lambda_{i_k}}\|x_{i_k} - \hat{x}_{k-1}\|^2 \\ &\le f(x_{i_k}) - \tilde{\ell}_f(x_{i_k}; \hat{x}_{k-1}) - \frac{1}{2\lambda_{i_k}}\|x_{i_k} - \hat{x}_{k-1}\|^2 \\ &\overset{(7)}{\le} 2M\|x_{i_k} - \hat{x}_{k-1}\| + \frac{L}{2}\|x_{i_k} - \hat{x}_{k-1}\|^2. \end{aligned} \tag{53}$$

Statement c) now follows from the above inequality, Assumption 4, and the fact that $x_{i_k}, \hat{x}_{k-1} \in \operatorname{dom} h$. ∎

It follows from Lemma 4.2(b) and the definition of $\underline{\lambda}$ in (42) that the overall number of bad iterations is

$$\mathcal{O}\left(\log_2\left((1-\tau)\left(\frac{M^2}{\varepsilon} + L\right)\right)\right).$$

4.1 Case where ϕ_* is known

This subsection considers the special case of Ad-GPB where ϕ_* is known and

$$\beta_0 = \frac{1}{2}, \qquad \hat{n}_k = \phi_* \quad \forall k \ge 1. \tag{54}$$

Even though the general result in Theorem 2.1 holds for this case, the simpler proof presented here covering the above case helps to understand the proof of the more general case given in Subsection 4.2 and has the advantage that it does not assume that $\operatorname{dom} h$ is bounded.

For convenience, the simplified version of Ad-GPB, referred to as Ad-GPB*, is explicitly stated below.

Ad-GPB*

---

0. Let $x_0 \in \operatorname{dom} h$, $\lambda_1 > 0$, $\tau \in (0,1)$, and $\varepsilon > 0$ be given; find $\Gamma_1 \in \mathcal{B}(\phi)$ such that $\Gamma_1 \ge \ell_\phi(\cdot; \hat{x}_0)$ and set $y_0 = \hat{x}_0$, $j_0 = 0$, $j = k = 1$;

1. compute x_j, m_j, y_j, and t_j as in (21), (22), (23), and (24), respectively;

2. **if** $t_j \le (\phi(y_j) - \phi_*)/2 + \varepsilon/4$ is violated **then** perform a **null update**, i.e.:
   set $\Gamma_{j+1} = \mathrm{BU}(\hat{x}_{k-1}, x_j, \Gamma_j, \lambda_j)$;
   if either $j = j_{k-1} + 1$ or

$$t_j - \tau t_{j-1} \le (1-\tau)\left[\frac{\phi(y_j) - \phi_*}{4} + \frac{\varepsilon}{8}\right], \tag{55}$$

   then set $\lambda_{j+1} = \lambda_j$; else, set $\lambda_{j+1} = \lambda_j/2$;
   else perform a **serious update**, i.e.:

set $\lambda_{j+1} = \lambda_j$ and find $\Gamma_{j+1} \in \mathcal{B}(\phi)$ such that $\Gamma_{j+1} \geq \ell_\phi(\cdot; x_j)$;
set $j_k = j$ and $(\hat{\lambda}_k, \hat{x}_k, \hat{y}_k, \hat{\Gamma}_k, \hat{m}_k, \hat{t}_k) = (\lambda_j, x_j, y_j, \Gamma_j, m_j, t_j)$;
if $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$, then output $(\hat{x}_k, \hat{y}_k)$, and **stop**;
$k \leftarrow k + 1$;

3. set $j \leftarrow j + 1$ and go to step 1.

We now make some remarks about Ad-GPB*. First, even though the parameter β_0 can be arbitrarily chosen in $(0, 1/2]$, Ad-GPB* is stated with $\beta_0 = 1/2$ for simplicity. Second, if Ad-GPB* reaches step 2 then the primal gap $\phi(y_j) - \phi_*$ is greater than ε because of step 2 and is substantially larger than this lower bound at its early cycles. Hence, the right-hand side $(\phi(y_j) - \phi_*)/2 + \varepsilon/4$ of its cycle termination criterion in step 2 is always larger than $3\varepsilon/4$ and is substantially larger than $3\varepsilon/4$ at its early cycles. Since, in contrast, GPB terminates a cycle when the inequality $t_j \leq \varepsilon/2$ is satisfied, the cycle termination of Ad-GPB* is always looser, and potentially much looser at its early cycles, than that of GPB.

The next lemma formally shows that Ad-GPB* is a specific instance of Ad-GPB.

Lemma 4.3 *The following statements hold:*

a) Ad-GPB is a special instance of Ad-GPB with β_0 and $\{\hat{n}_k\}$ chosen as in (54); moreover, $\beta_k = 1/2$ for every cycle index $k \geq 1$;*

b) for every cycle index k of Ad-GPB, we have*

$$\hat{t}_k \leq [\phi(\hat{y}_k) - \phi_*]/2 + \varepsilon/4. \tag{56}$$

Proof: a) The first claim of (a) is obvious. To show that $\beta_k = 1/2$ for every index cycle $k \geq 1$ generated by Ad-GPB, it suffices to show that $\beta_k = \beta_{k-1}$ because $\beta_0 = 1/2$. Indeed, using (54), the facts that $\beta_l \leq \beta_0 = 1/2$ for $l \geq 0$ due to Lemma 4.1(c), and the definitions of ϕ_k^a and $\hat{g}_k$ in (28) and (27), respectively, we conclude that

$$\hat{g}_k \leq \frac{1}{2\sum_{l=\lceil k/2 \rceil}^{k} \hat{\lambda}_l} \sum_{l=\lceil k/2 \rceil}^{k} \hat{\lambda}_l [\phi(\hat{y}_l) - \phi_*] = \frac{\hat{\phi}_k^a - \phi_*}{2},$$

and hence that $\beta_k = \beta_{k-1}$ due to the update rule for β_k at the end of step 2 of Ad-GPB.

b) This statement follows from (a), Lemma 4.1(b), and the fact that $\beta_0 = 1/2$. ∎

Let d_0 denote the distance of the initial point $\hat{x}_0 \in \operatorname{dom} h$ to the set of optimal solutions X^*, i.e.,

$$d_0 := \|\hat{x}_0 - \hat{x}_*\|, \quad \text{where} \quad \hat{x}_* := \operatorname{argmin} \{\|\hat{x}_0 - x_*\| : x_* \in X_*\}. \tag{57}$$

Lemma 4.4 *If $K \geq 1$ is a cycle index generated by Ad-GPB, then we have*

$$\sum_{k=1}^{K} \hat{\lambda}_k [\phi(\hat{y}_k) - \phi_*] \leq \frac{\varepsilon}{2} \sum_{k=1}^{K} \hat{\lambda}_k + d_0^2. \tag{58}$$

Proof: Relation (50) with $u = \hat{x}_*$, and the facts that $\hat{\Gamma}_k \leq \phi$ and $\phi_* = \phi(\hat{x}_*)$, imply that

$$\begin{aligned}
\hat{\lambda}_k [\phi(\hat{y}_k) - \phi_*] &\leq \hat{\lambda}_k [\phi(\hat{y}_k) - \hat{\Gamma}_k(\hat{x}_*)] \\
&\leq \hat{\lambda}_k \hat{t}_k + \frac{1}{2}\|\hat{x}_* - \hat{x}_{k-1}\|^2 - \frac{1}{2}\|\hat{x}_k - \hat{x}_*\|^2 \\
&\leq \hat{\lambda}_k \left[\frac{\phi(\hat{y}_k) - \phi_*}{2} + \frac{\varepsilon}{4}\right] + \frac{1}{2}\|\hat{x}_* - \hat{x}_{k-1}\|^2 - \frac{1}{2}\|\hat{x}_k - \hat{x}_*\|^2
\end{aligned}$$

where the last inequality is due to (56). Simplifying the above inequality and summing the resulting inequality from $k = 1, \ldots, K$, we conclude that

$$\frac{1}{2}\sum_{k=1}^{K} \hat{\lambda}_k [\phi(\hat{y}_k) - \phi_*] \leq \frac{\varepsilon}{4} \sum_{k=1}^{K} \hat{\lambda}_k + \frac{1}{2}\|\hat{x}_* - \hat{x}_0\|^2 - \frac{1}{2}\|\hat{x}_K - \hat{x}_*\|^2.$$

The statement now follows from the above inequality and (57). ∎

We are now ready to prove Theorem 4.5.

Theorem 4.5 *Ad-GPB* finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$ in at most $\hat{K}(\varepsilon)$ cycles and*

$$\log_2^+ \left(\bar{Q}\lambda_1 \left(\frac{M^2}{\varepsilon} + \frac{L}{16} \right) \right) + \left(\frac{1+\tau}{1-\tau} \log^+ \left[8\bar{t}\varepsilon^{-1} \right] + 2 \right) \hat{K}(\varepsilon) \tag{59}$$

iterations, where $\bar{Q}$ is as in (35) *and*

$$\hat{K}(\varepsilon) := \left\lceil \frac{2d_0^2\bar{Q}}{\varepsilon} \left(\frac{M^2}{\varepsilon} + \frac{L}{16} + \frac{1}{\lambda_1 \bar{Q}} \right) \right\rceil. \tag{60}$$

Proof: We first prove that Ad-GPB finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$ in at most $\hat{K}(\varepsilon)$ cycles. Suppose for contradiction that Ad-GPB generates a cycle $K > \hat{K}(\varepsilon)$. Since the Ad-GPB did not stop at any of the previous iterations, we have that $\phi(\hat{y}_k) - \phi_* > \varepsilon$ for every $k = 1, \ldots, K-1$. Using the previous observation, inequality (58), the fact that $K - 1 \geq \hat{K}(\varepsilon)$, and Lemma 4.2(a), we conclude that

$$\begin{aligned} \frac{d_0^2}{\varepsilon} &\overset{(58)}{\geq} \frac{1}{\varepsilon} \sum_{k=1}^{K-1} \hat{\lambda}_k [\phi(\hat{y}_k) - \phi_*] - \frac{1}{2} \sum_{k=1}^{K-1} \hat{\lambda}_k > \sum_{k=1}^{K-1} \hat{\lambda}_k - \frac{1}{2} \sum_{k=1}^{K-1} \hat{\lambda}_k \\ &\geq \frac{1}{2} \min \{\underline{\lambda}, \lambda_1\} (K-1) \geq \frac{1}{2} \min \{\underline{\lambda}, \lambda_1\} \hat{K}(\varepsilon). \end{aligned}$$

The definition of $\hat{K}(\varepsilon)$ in (60), the above inequality, and some simple algebraic manipulation on the min term, yield the desired contradiction.

Let $\bar{k} \leq \hat{K}(\varepsilon)$ denote the numbers of cycles generated by Ad-GPB*. Proposition 3.5, and statements (b) and (c) of Lemma 4.2, then imply that the total number of iterations performed by Ad-GPB* until it finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$ is bounded by

$$\begin{aligned} \sum_{k=1}^{\bar{k}} |\mathcal{C}_k| &\leq \sum_{k=1}^{\bar{k}} \left(\frac{1+\tau}{1-\tau} \log^+ \left[8\bar{t}\varepsilon^{-1} \right] + 2 + s_k \right) \\ &\leq \log_2^+ \frac{\lambda_1}{\underline{\lambda}} + \left(\frac{1+\tau}{1-\tau} \log^+ \left[8\bar{t}\varepsilon^{-1} \right] + 2 \right) \hat{K}(\varepsilon) \end{aligned}$$

and hence by (59), due to the definitions of $\bar{Q}$ and $\underline{\lambda}$ in (35) and (42), respectively. ∎

4.2 General case where ϕ_* is unknown

As already mentioned above, this subsection considers the general case (see step 2 of Ad-GPB) where $\hat{n}_k$ is in the interval $[\hat{\ell}_k, \phi_*]$, where $\hat{\ell}_k$ is as in (26), and derives an upper bound on the number of cycles generated by Ad-GPB.

Lemma 4.6 *For every cycle index k of Ad-GPB, we have:*

$$\hat{\phi}_k^a - \hat{n}_k \leq \hat{g}_k + \frac{D^2}{2\sum_{l=\lceil k/2 \rceil}^k \hat{\lambda}_l} + \frac{\varepsilon}{4} \tag{61}$$

where D is as in (4).

Proof: Let $u \in \operatorname{dom} h$ be given. Multiplying (50) by $\hat{\lambda}_k$ and summing the resulting inequality from $k = \lceil k/2 \rceil, \ldots, k$, we have

$$\begin{aligned} \sum_{l=\lceil k/2 \rceil}^k \hat{\lambda}_l [\phi(\hat{y}_l) - \hat{\Gamma}_l(u)] &\leq \sum_{l=\lceil k/2 \rceil}^k \left(\hat{\lambda}_l \hat{t}_l + \frac{1}{2} \left[\|u - \hat{x}_{l-1}\|^2 - \|u - \hat{x}_l\|^2 \right] \right) \\ &\leq \sum_{l=\lceil k/2 \rceil}^k \hat{\lambda}_l \left[\beta_{l-1} [\phi(\hat{y}_l) - \hat{n}_{l-1}] + \frac{\varepsilon}{4} \right] + \frac{1}{2} \|u - \hat{x}_{\lceil k/2 \rceil - 1}\|^2 - \frac{1}{2} \|\hat{x}_k - u\|^2, \end{aligned}$$

where the last inequality is due to Lemma 4.1(b). Dividing both sides of the above inequality by $\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l$, and using the definition of $\hat{\phi}_k^a$, $\hat{\Gamma}_k^a(\cdot)$, and $\hat{g}_k$ in (27), (26), and (28), respectively, we have

$$\hat{\phi}_k^a - \hat{\Gamma}_k^a(u) \le \hat{g}_k + \frac{\varepsilon}{4} + \frac{\|u-\hat{x}_{\lceil k/2\rceil-1}\|^2}{2\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l} \le \hat{g}_k + \frac{\varepsilon}{4} + \frac{D^2}{2\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l},$$

where the last inequality is due to Assumption (A4) and the fact that $u \in \operatorname{dom} h$ and $\hat{x}_{\lceil k/2\rceil-1} \in \operatorname{dom} h$ where the last inclusion is due to Lemma 4.1(c). Inequality (61) now follows from (31) and by maximizing the above inequality relative to $u \in \operatorname{dom} h$. ∎

Lemma 4.7 *For every cycle index k of Ad-GPB, the following statements hold:*

a) If $\beta_k = \beta_{k-1}$, then we have

$$\hat{\phi}_k^a - \hat{n}_k \le \frac{D^2}{\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l} + \frac{\varepsilon}{2}; \tag{62}$$

b) If $\beta_k = \beta_{k-1}/2$, then we have

$$\frac{\hat{\phi}_k^a - \hat{n}_k}{2} < \hat{g}_k \le \beta_{\lceil \frac{k}{2}\rceil-1}(\phi(\hat{x}_0) - \hat{n}_0) \tag{63}$$

where $\hat{\phi}_k^a$ is as in (27).

Proof: a) The update rule for β_k just after equation (28) and the assumption that $\beta_k = \beta_{k-1}$ imply that $\hat{g}_k \le (\hat{\phi}_k^a - \hat{n}_k)/2$. This observation and inequality (61) then immediately imply (62).

b) The first inequality in (63) follows from the assumption that $\beta_k = \beta_{k-1}/2$ and the update rule for β_k just after equation (28). Moreover, using the definition of $\hat{g}_k$ in (28), the fact that $\beta_{l-1} \le \beta_{\lceil k/2\rceil-1}$ for every $l \ge \lceil k/2\rceil$ due to Lemma 4.1(d), and the fact that $\hat{n}_k \ge \hat{\ell}_k$ for every $k \ge 1$ in (31) we conclude that

$$\hat{g}_k \le \frac{\beta_{\lceil k/2\rceil-1}}{\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l}\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l[\phi(\hat{y}_l) - \hat{n}_{l-1}] \le \frac{\beta_{\lceil k/2\rceil-1}}{\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l}\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l[\phi(\hat{y}_l) - \hat{\ell}_{l-1}] \le \beta_{\lceil k/2\rceil-1}[\phi(\hat{y}_0) - \hat{\ell}_0],$$

where the last inequality is because statements (c) and (d) of Lemma 4.1 imply that $\phi(\hat{y}_l) \le \phi(\hat{y}_0)$ and $\hat{\ell}_l \ge \hat{\ell}_0$ for every $l \ge 1$. The above inequality, the convention that $\hat{y}_0 = \hat{x}_0$, and the fact that $\hat{n}_0 = \hat{\ell}_0$ (see step 0 of Ad-GPB) then imply the second inequality in (63). ∎

We are now ready to prove the main result of this paper.

Proof of Theorem 2.1 To simplify notation, let $\bar{K} = \bar{K}(\varepsilon)$. It is easy to see that

$$\frac{D^2}{\bar{K}}\left(\frac{1}{\underline{\lambda}} + \frac{1}{\lambda_1}\right) \le \frac{\varepsilon}{4}, \qquad \frac{1}{2^{\bar{K}-2}}(\phi(\hat{x}_0) - \hat{n}_0) < \frac{\varepsilon}{\beta_0}. \tag{64}$$

We first prove that Ad-GPB finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \hat{n}_k \le \varepsilon$ in at most $4\bar{K}$ cycles. Suppose for contradiction that Ad-GPB generates a cycle $K \ge 4\bar{K}+1$. Since the Ad-GPB did not stop at cycles from 1 to $K-1$, we have that

$$\phi(\hat{y}_k) - \hat{n}_k > \varepsilon \quad \forall k \in \{1,\dots,4\bar{K}\}. \tag{65}$$

We then have that

$$\beta_k = \frac{\beta_{k-1}}{2} \quad \forall k \in \{\bar{K},\dots,4\bar{K}\} \tag{66}$$

since otherwise we would have some $\beta_k = \beta_{k-1}$ for some $k \in \{\bar{K},\dots,4\bar{K}\}$, and this together with (65), Lemma 4.1(c), Lemma 4.7(a), and Lemma 4.2(a), would yield the contradiction that

$$\begin{aligned}\varepsilon < \phi(\hat{y}_k) - \hat{n}_k &\overset{\text{L.4.1}(c)}{\le} \hat{\phi}_k^a - \hat{n}_k \overset{\text{L.4.7}(a)}{\le} \frac{D^2}{\sum_{l=\lceil k/2\rceil}^{k}\hat{\lambda}_l} + \frac{\varepsilon}{2} \overset{\text{L.4.2}(a)}{\le} \frac{2D^2}{k\min\{\underline{\lambda},\lambda_1\}} + \frac{\varepsilon}{2} \\ &\le \frac{2D^2}{\bar{K}}\left(\frac{1}{\underline{\lambda}} + \frac{1}{\lambda_1}\right) + \frac{\varepsilon}{2} \overset{(64)}{\le} \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon,\end{aligned}$$

where the last inequality is due to the first inequality in (64). Relations (65) and (66), and Lemmas 4.1(c) and 4.7(b), all with $k = 4\bar{K}$, then yield

$$\varepsilon \overset{(65)}{<} \phi(\hat{y}_{4\bar{K}}) - \hat{n}_{4\bar{K}} \overset{\mathbf{L}.4.1(c)}{\leq} \hat{\phi}^a_{4\bar{K}} - \hat{n}_{4\bar{K}} \overset{\mathbf{L}.4.7(b)}{\leq} 2\beta_{2\bar{K}-1}(\phi(\hat{x}_0) - \hat{n}_0) \overset{(66)}{\leq} \frac{1}{2^{\bar{K}-2}}\beta_{\bar{K}}(\phi(\hat{x}_0) - \hat{n}_0) \overset{(64)}{<} \frac{\beta_{\bar{K}}}{\beta_0}\varepsilon$$

where the last inequality is due to the second inequality in (64). Since $\beta_{\bar{K}} \leq \beta_0$, the above inequality gives the desired contradiction, and hence the first conclusion of theorem holds.

To show the second conclusion of the theorem, let $\bar{k} \leq 4\bar{K}$ denote the numbers of cycles generated by Ad-GPB. Proposition 3.5, and statements (b) and (c) of Lemma 4.2 then imply that the total number of iterations that Ad-GPB finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \hat{n}_k \leq \varepsilon$ is bounded by

$$\begin{aligned}\sum_{k=1}^{\bar{k}} |\mathcal{C}_k| &\leq \sum_{k=1}^{\bar{k}} \left(\frac{1+\tau}{1-\tau} \log^+ \left[8\bar{t}\bar{\varepsilon}^{-1} \right] + 2 + s_k \right) \\ &\leq \log_2^+ \frac{\lambda_1}{\underline{\lambda}} + 4\bar{K} \left(\frac{1+\tau}{1-\tau} \log^+ \left[8\bar{t}\bar{\varepsilon}^{-1} \right] + 2 \right)\end{aligned}$$

and hence by (36), due to the definitions of $\bar{Q}$ and $\underline{\lambda}$ in (35) and (42), respectively. ∎

5 Computational experiments

This section reports the computational results of Ad-GPB* and two corresponding practical variants against other modern PB methods and the subgradient method. It contains two subsections. The first one presents the computational results for a simple l_1 feasibility problem. The second one showcases the computational results for the Lagrangian cut problem appeared in the area of integer programming.

All the methods tested in the following two subsections are terminated based on the following criterion:

$$\phi(x_k) - \phi_* \leq \bar{\varepsilon}[\phi(x_0) - \phi_*] \tag{67}$$

where $\bar{\varepsilon} = 10^{-6}$, 10^{-5} or 10^{-4}. All experiments were performed in MATLAB 2023a and run on a PC with a 16-core Intel Core i9 processor and 32 GB of memory.

Now we describe the algorithm details used in the following two subsections. We first describe Polyak subgradient method. Given x_k, it computes

$$x_{k+1} = \operatorname{argmin}_x \left\{ \ell_\phi(x; x_k) + \frac{1}{2\lambda_{\text{pol}}(x_k)} \|x - x_k\|^2 \right\}$$

where

$$\lambda_{\text{pol}}(x) := \frac{\phi(x) - \phi_*}{\|g(x)\|^2} \tag{68}$$

where $g(x) \in \partial f(x)$. Next we describe five GPB related methods, namely: GPB, Ad-GPB*, Ad-GPB**, Pol-Ad-GPB*, and Pol-GPB. A cycle $k \geq 1$ of Ad-GPB*, regardless of the way its initial prox stepsize is chosen, is called good if the prox stepsizes λ_j do not change (i.e., the inequality (25) is not violated) within it. First, GPB is stated in [18, 19]. Second, Ad-GPB* is stated in Subsection 4.1. Third, Ad-GPB** is a corresponding variant of Ad-GPB* that allows the prox stepsize to increase at the beginning of its initial cycles. Specifically, if $\bar{k}$ denotes the largest cycle index for which cycles 1 to $\bar{k}$ are good then Ad-GPB** sets $\lambda_{i_{k+1}} = 2\hat{\lambda}_k$ for every $k \leq \bar{k}$ and afterwards sets $\lambda_{i_{k+1}} = \hat{\lambda}_k$ for every $k > \bar{k}$ as Ad-GPB* does, where $\hat{\lambda}_k$ and i_{k+1} are defined in step 2 of Ad-GPB* and (29), respectively. The motivation behind Ad-GPB** is to prevent Ad-GPB or Ad-GPB* from generating only small λ_j's due to a poor choice of initial prox stepsize λ_1. Pol-GPB and Pol-Ad-GPB* are two Polyak-type variants of GPB and Ad-GPB* where the initial prox stepsize for the kth-cycle is set to $\lambda_{i_k} = 40\lambda_{\text{pol}}(\hat{x}_{k-1})$. The above five methods all update the bundle Γ according to the 2-cut update scheme described in Subsection 2.2. Finally, Ad-GPB*, Ad-GPB** and Pol-Ad-GPB* use $\tau = 0.95$.

5.1 l_1 feasibility problem

This subsection reports the computational results on a l_1 feasibility problem. It contains two subsections. The first one presents computational results of Ad-GPB* and Ad-GPB** against GPB method in [18, 19]. The second one showcases the computational results of Pol-Ad-GPB* and Pol-GPB against subgradient method with Polyak stepsize.

We start by describing the l_1 feasibility problem. The problem can be formulated as:

$$\phi_* := \min_{x \geq 0} f(x) := \|Ax - b\|_1 \tag{69}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^n$ are known. We consider two different ways of generating the data, i.e., sparse and dense ways. For dense problems, matrix A is randomly generated in the form $A = NU$ where the entries of the matrix $N \in \mathbb{R}^{m \times n}$ (resp., $U \in \mathbb{R}^{n \times n}$) are i.i.d sampled from the standard normal $\mathcal{N}(0,1)$ (resp., uniform $\mathcal{U}[0,100]$) distribution. For sparse problem, matrix A is randomly generated in the form $A = DN$ where the nonzero entries of the sparse matrix $N \in \mathbb{R}^{m \times n}$ are i.i.d sampled from the standard normal $\mathcal{N}(0,1)$ distribution and D is a diagonal matrix where the diagonal of D are i.i.d sampled from $\mathcal{U}[0,1000]$. In both cases, vector b is determined as $b = Ax_*$ where $x_* = (v_*)^2$ for some vector $v_* \in \mathbb{R}^n$ whose entries are i.i.d sampled from the standard Normal distribution $\mathcal{N}(0,1)$. Finally, we generated $x_0 = (v_0)^2$ for some vector $v_0 \in \mathbb{R}^n$ whose entries are i.i.d. sampled from the uniform distribution over $(0,1)$. Clearly, x_* is a global minimizer of (69), whose optimal value f_* equals zero in both cases. We test our methods on six dense and six sparse instances.

We now describe some details about all the tables that appear in this subsection. We set the target $\bar{\varepsilon}$ in (67) as 10^{-5} for dense instances and 10^{-4} for sparse instances. The quantities θ_m, θ_n and θ_s are defined as $\theta_m = m/10^3$, $\theta_n = n/10^3$, and $\theta_s := \text{nnz}(A)/mn$, where $\text{nnz}(A)$ is the number of non-zero entries of A. An entry in each table is given as a fraction with the numerator expressing the (rounded) number of iterations and the denominator expressing the CPU running time in seconds. An entry marked as $*/*$ indicates that the CPU running time exceeds the allocated time limit. The bold numbers highlight the method that has the best performance for each instance.

5.1.1 Ad-GPB* versus GPB

This subsection presents computational results of GPB method in [18, 19] against Ad-GPB* and Ad-GPB**.

To check how sensitive the three methods are relative to the initial choice of prox stepsize λ_1, we test them for $\lambda_1 = \alpha\lambda_{\text{pol}}(x_0)$ where $\alpha \in \{0.01, 1, 100\}$. The computational results for the above three methods are given in Table 1 (resp., Table 2) for six sparse (resp., dense) instances. The time limit is four hours for Table 1 and two hours for Table 2.

ALG.	GPB			Ad-GPB*			Ad-GPB**		
$(\theta_m, \theta_n, \theta_s)$ \ α	10^{-2}	1	10^2	10^{-2}	1	10^2	10^{-2}	1	10^2
(1,20,10^{-2})	$\frac{68.3K}{125}$	$\frac{153.7K}{314}$	$\frac{*}{*}$	$\frac{26.1K}{36}$	$\frac{19.6K}{27}$	$\frac{23.3K}{33}$	$\mathbf{\frac{17.8K}{26}}$	$\frac{21.3K}{31}$	$\frac{23.3K}{34}$
(3,30,10^{-2})	$\frac{164.2K}{450}$	$\frac{120.8K}{381}$	$\frac{*}{*}$	$\frac{59.8K}{152}$	$\frac{39.5k}{102}$	$\frac{62.4K}{164}$	$\frac{55.4K}{144}$	$\mathbf{\frac{36.3K}{94}}$	$\frac{62.4K}{165}$
(5,50,10^{-2})	$\frac{123.4K}{811}$	$\frac{99.4K}{654}$	$\frac{*}{*}$	$\frac{62.8K}{409}$	$\frac{32.7K}{212}$	$\frac{64.5K}{409}$	$\frac{59.1K}{387}$	$\mathbf{\frac{32.1K}{211}}$	$\frac{64.5K}{420}$
(10,100,10^{-3})	$\frac{152.2K}{928}$	$\frac{132.0K}{837}$	$\frac{*}{*}$	$\frac{67.4K}{363}$	$\mathbf{\frac{40.5K}{226}}$	$\frac{66.9K}{360}$	$\frac{61.8K}{337}$	$\frac{44.4K}{242}$	$\frac{66.9K}{360}$
(20,200,10^{-3})	$\frac{136.2K}{7119}$	$\frac{111.2K}{5725}$	$\frac{*}{*}$	$\frac{78.4K}{2636}$	$\frac{41.3K}{1401}$	$\frac{76.0K}{2577}$	$\frac{67.9K}{2384}$	$\mathbf{\frac{40.8K}{1441}}$	$\frac{76.0K}{3280}$
(50,500,10^{-4})	$\frac{148.1K}{12574}$	$\frac{130.1K}{11864}$	$\frac{*}{*}$	$\frac{70.4K}{3947}$	$\mathbf{\frac{42.9K}{2460}}$	$\frac{65.0K}{3803}$	$\frac{67.1K}{3820}$	$\frac{43.0K}{2392}$	$\frac{65.0K}{3610}$

Table 1: Numerical results for sparse instances. A relative tolerance of $\bar{\varepsilon} = 10^{-4}$ is set and a time limit of 14400 seconds (4 hours) is given.

ALG.	GPB			Ad-GPB*			Ad-GPB**		
(θ_m, θ_n) \ α	10^{-2}	1	10^2	10^{-2}	1	10^2	10^{-2}	1	10^2
(0.5,1.5)	$\frac{9354.7K}{1502}$	$\frac{3323.6K}{462}$	$\frac{*}{*}$	$\frac{74.2K}{6}$	$\frac{\mathbf{47.9K}}{\mathbf{5}}$	$\frac{49.7K}{5}$	$\frac{53.8K}{5}$	$\frac{56.4K}{6}$	$\frac{49.7K}{5}$
(1,3)	$\frac{*}{*}$	$\frac{5384.9K}{3114}$	$\frac{*}{*}$	$\frac{86.7K}{42}$	$\frac{81.9K}{40}$	$\frac{87.0K}{43}$	$\frac{137.3K}{70}$	$\frac{\mathbf{79.0K}}{\mathbf{41}}$	$\frac{143.1K}{75}$
(2,6)	$\frac{*}{*}$	$\frac{221.8K}{1214}$	$\frac{\mathbf{59.0K}}{\mathbf{509}}$	$\frac{305.6K}{1552}$	$\frac{181.4K}{911}$	$\frac{134.7K}{677}$	$\frac{136.7K}{685}$	$\frac{176.6K}{882}$	$\frac{133.9K}{669}$
(1.5,0.5)	$\frac{1630.7K}{181}$	$\frac{495.1K}{55}$	$\frac{*}{*}$	$\frac{135.5K}{13}$	$\frac{\mathbf{102.5K}}{\mathbf{10}}$	$\frac{113.1K}{11}$	$\frac{128.4K}{13}$	$\frac{104.6K}{10}$	$\frac{117.8K}{11}$
(3,1)	$\frac{2170.7K}{869}$	$\frac{502.5K}{199}$	$\frac{*}{*}$	$\frac{233.5K}{100}$	$\frac{\mathbf{155.5K}}{\mathbf{65}}$	$\frac{166.5K}{74}$	$\frac{180.8K}{71}$	$\frac{156.7K}{64}$	$\frac{175.5K}{73}$
(6,2)	$\frac{*}{*}$	$\frac{757.9K}{3542}$	$\frac{*}{*}$	$\frac{351.4K}{1779}$	$\frac{242.4K}{1211}$	$\frac{276.6K}{1376}$	$\frac{304.0K}{1515}$	$\frac{\mathbf{238.0K}}{\mathbf{1151}}$	$\frac{276.6K}{1340}$

Table 2: Numerical results for dense instances. A relative tolerance of $\bar{\varepsilon} = 10^{-5}$ is set and a time limit of 7200 seconds (2 hours) is given.

The results in Tables 1 and 2 show that Ad-GPB* and Ad-GPB** are generally at least two to three times faster than GPB in terms of CPU running time. Second, it also shows that Ad-GPB* and Ad-GPB** are more robust to initial stepsize than GPB. We also observe that Ad-GPB** generally performs slightly better for small initial stepsize than Ad-GPB* which accounts for the increase of stepsize at the end of the cycle.

5.1.2 Polyak type methods

This subsection considers two Polyak-type variants of GPB and Ad-GPB* where the initial prox stepsize for the kth-cycle is set to $\lambda_{i_k} = 40\lambda_{\text{pol}}(\hat{x}_{k-1})$. These two variants in turn are compared with the subgradient method with Polyak stepsize (see (68)) and the Ad-GPB* and Ad-GPB** variants described in Subsection 5.1.1.

The computational results for the above five methods are given in Table 3 (resp., Table 4) for six sparse (resp., dense) instances. The results for Ad-GPB* and Ad-GPB** are the same ones that appear in Tables 1 and 2 with $\alpha = 1$. They are duplicated here for the sake of convenience.

$(\theta_m, \theta_n, \theta_s)$	Pol-Sub	Pol-GPB	Pol-Ad-GPB*	Ad-GPB*	Ad-GPB**
(1,20,10^{-2})	431.3K/354	33.8K/58	**15.5K/22**	19.6K/27	21.3K/31
(3,30,10^{-2})	413.3K/786	126.9K/392	**21.2K/60**	39.5K/102	36.3K/94
(5,50,10^{-2})	389.8k/2440	91.4K/581	**19.7K/128**	32.7K/212	32.1K/211
(10,100,10^{-3})	473.2k/2092	137.12K/876	**21.3K/157**	40.5K/226	44.4K/242
(20,200,10^{-3})	*/*	115.7K/5576	**25.9K/1070**	41.3K/1401	40.8K/1441
(50,500,10^{-4})	*/*	133.0K/13754	**25.5K/1847**	42.9K/2460	43.0K/2392

Table 3: Numerical results for sparse instances. A relative tolerance of $\bar{\varepsilon} = 10^{-4}$ is set and a time limit of 14400 seconds (4 hours) is given.

(θ_m, θ_n)	Pol-Sub	Pol-GPB	Pol-Ad-GPB*	Ad-GPB*	Ad-GPB**
(0.5,1.5)	479.8K/36.5	143.2K/22.8	73.5K/9.4	47.9K/5	**56.4K/6**
(1,3)	1644.3K/1440	390.3K/196.7	144.4K/72.1	81.9K/40	**79.0K/41**
(2,6)	354.5K/2513	**46.4K/233**	182.5K/959	181.4K/911	176.6K/882
(1.5,0.5)	699.4K/79.5	109.9K/12.0	**56.5K/5.6**	102.5K/10	104.6K/10
(3,1)	1034.6K/1046	147.8K/57.9	**89.2K/38.8**	155.5K/65	156.7K/64
(6,2)	*/*	**136.1K/602**	143.1K/713	242.4K/1211	238.0K/1151

Table 4: Numerical results for dense instances. A relative tolerance of $\bar{\varepsilon} = 10^{-5}$ is set and a time limit of 7200 seconds (2 hours) is given.

Tables 3 and 4 demonstrate that PB methods generally outperform Pol-Subgrad in terms of CPU running time. Additionally, Pol-Ad-GPB* stands out as a particularly effective variant, outperforming other methods in eight out of twelve instances.

5.2 Lagrangian cut problem

This subsection presents the numerical results comparing Ad-GPB* and Pol-Ad-GPB* against GPB and Pol-Sub on a convex nonsmooth optimization problem that has broad applications in the field of integer programming (see e.g. [28]).

The problem considered in this subsection arises in the context of solving the the stochastic binary multi-knapsack problem

$$\begin{array}{ll} \min & c^T x + P(x) \\ \text{s.t.} & Ax \geq b \\ & x \in \{0,1\}^n \end{array} \tag{70}$$

where $P(x) := \mathbb{E}_\xi \left[P_\xi(x) \right]$ and

$$\begin{array}{rl} P_\xi(x) := \min & q(\xi)^T y \\ \text{s.t.} & Wy \geq h - Tx \\ & y \in \{0,1\}^n \end{array} \tag{71}$$

for every $x \in \{0,1\}^n$. In the second-stage problem, only the objective vector $q(\xi)$ is a random variable. Moreover, it is assumed that its support Ξ is a finite set, i.e., $q(\cdot)$ has a finite number of scenarios ξ's.

Benders decomposition (see e.g. [7, 26]) is an efficient cutting-plane approach for solving (70) which approximates $P(\cdot)$ by pointwise maximum of cuts for $P(\cdot)$. Specifically, an affine function $A(\cdot)$ such that $P(x') \geq \mathcal{A}(x')$ for every $x' \in \{0,1\}^n$ is called a cut for $P(\cdot)$; moreover, a cut for $P(\cdot)$ is tight at x if $P(x) = \mathcal{A}(x)$. Benders decomposition starts with a cut $\mathcal{A}_0$ for $P(\cdot)$ and compute a sequence $\{x_k\}$ of iterates as follows: given cuts $\{\mathcal{A}_i(\cdot)\}_{i=0}^{k-1}$ for $P(\cdot)$, it computes x_k as

$$\begin{array}{rl} x_k = \operatorname{argmin}_x & c^T x + P_k(x) \\ \text{s.t.} & Ax \geq b \\ & x \in \{0,1\}^n \end{array} \tag{72}$$

where

$$P_k(\cdot) = \max_{i=0,\cdots,k-1} \mathcal{A}_i(\cdot);$$

it then uses x_k to generate a new cut $\mathcal{A}_k$ for $P(\cdot)$ and repeats the above steps with k replaced by $k+1$. Problem (72) can be easily formulated as an equivalent linear integer programming problem.

We now describe how to generate a Lagrangian cut for $P(\cdot)$ using a given point $x \in \{0,1\}^n$. First, for every $\xi \in \Xi$, (71) is equivalent to

$$\begin{aligned} \min_{y,u} \quad & q(\xi)^T y \\ \text{s.t.} \quad & Wy + Tu \geq h \\ & y \in \{0,1\}^n,\ u \in [0,1]^n \\ & u - x = 0. \end{aligned}$$

By dualizing the constraint $u - x = 0$, we obtain the Lagrangian dual (LD) problem

$$D_\xi(x) := \max_\pi L_\xi(x;\pi) \tag{73}$$

where

$$\begin{aligned} L_\xi(x;\pi) := \min_{y,u} \ & q(\xi)^T y - \pi^T(u - x) \\ \text{s.t.} \quad & Wy + Tu \geq d \\ & y \in \{0,1\}^n,\ u \in [0,1]^n. \end{aligned} \tag{74}$$

Let $\pi_\xi(x)$ denote an optimal solution of (73). The optimal values $P_\xi(\cdot)$ and $D_\xi(\cdot)$ of (71) and (73), respectively, are known to satisfy the following two properties for every $\xi \in \Xi$ and $x \in \{0,1\}^n$:

(i) $P_\xi(x') \geq D_\xi(x') \geq D_\xi(x) + \langle \pi_\xi(x), x' - x \rangle$ for every $x' \in \{0,1\}^n$;

(ii) $P_\xi(x) = D_\xi(x)$.

Property (i) can be found in many textbooks dealing with Lagrangian duality theory and property (ii) has been established in [28]. Defining

$$\pi(x) := \mathbb{E}[\pi_\xi(x)]$$

and taking expectation of the relations in (i) and (ii), we easily see that

$$P(x') \geq P(x) + \langle \pi(x), x' - x \rangle \quad \forall x' \in \{0,1\}^n,$$

and hence that $\mathcal{A}_x(\cdot) := P(x) + \langle \pi(x), \cdot - x \rangle$ is a tight cut for $P(\cdot)$ at x.

Computation of $P(x)$ assumes that the optimal value $P_\xi(\cdot)$ of (71) can be efficiently computed for every $\xi \in \Xi$. Computation of $\pi(x)$ assumes that an optimal solution of (73) can be computed for every $\xi \in \Xi$. Noting that (73) is an unconstrained convex nonsmooth optimization problem in terms of variable π and its optimal value $D_\xi(x)$ is the (already computed) optimal value $P_\xi(x)$ of (71), we use the Ad-GPB* variant of Ad-GPB to obtain a near optimal solution $\approx \pi_\xi(x)$ of (73). For the purpose of this subsection, we use several instances of (73) to benchmark the methods described at the beginning of this section.

For every $(\xi, x) \in \Xi \times \{0,1\}^n$, recall that using Ad-GPB* to solve (73) requires the ability to evaluate $L_\xi(x,\cdot)$ and compute a subgradient of $-L_\xi(x,\cdot)$ at every $\pi \in \mathbb{R}^n$. The value $L_\xi(x,\pi)$ is evaluated by solving MILP (74). Moreover, if $(u_\xi(x;\pi), y_\xi(x;\pi)))$ denotes an optimal solution of (74), then $u_\xi(x;\pi)$ yields a subgradient of $-L_\xi(x,\cdot)$ at π. It is worth noting (73) is a non-smooth convex problem that does not seem to be tractable by the methods discussed in the papers (see e.g. [3, 9, 10, 14, 22, 23, 24, 25]) for solving min-max smooth convex-concave saddle-point problems, mainly due to the integrality condition imposed on the decision variable y in (74).

Next we describe how the data of (70) and (71) is generated. We generate three random instances of (70), each following the same methodology as in [1]. We set $n = 240$ and $\Xi = \{1, \ldots, 20\}$ with each scenario $\xi \in \Xi$ being equiprobable. We generate matrices $A_1, A_2 \in \mathbb{R}^{50\times 120}$, $T_1, W \in \mathbb{R}^{5\times 120}$, and vector $c \in \mathbb{R}^{240}$, with all entries i.i.d. sampled from the uniform distribution over the integers $\{1, \ldots, 100\}$. We then set $A = [A_1\ A_2]$ and $T = [T_1\ 0]$ where the zero block of T is 5×120. Twenty vectors $\{q(\xi)\}_{\xi\in\Xi}$ with components i.i.d. sampled from $\{1, \ldots, 100\}$ are generated. Finally, we set $b = 3(A_1\mathbf{1} + A_2\mathbf{1})/4$ and $h = 3(W\mathbf{1} + T_1\mathbf{1})/4$ where $\mathbf{1}$ denotes the vector of ones.

For each randomly generated instance of (70), we run Benders decomposition started from $x_0 = \mathbf{1}$ to obtain three iterates x_1, x_2, and x_3. Each $P(x_k)$ and $\pi(x_k)$ for $k = 1, 2, 3$ are computed using the

twenty randomly generated vectors $\{q(\xi)\}_{\xi\in\Xi}$ as described above, and hence each iteration solves twenty LD subproblems as in (73). Hence, each randomly generated instance of (70) yields a total of sixty LD instances as in (73). The total time to solve these sixty LD instances are given in Table 5 for the three instances of (70) (named $I1$, $I2$ and $I3$ in the table) and all the benchmarked methods considered in this section. In this comparison, both Ad-GPB* and GPB set the initial stepsize λ_1 to $\lambda_{\rm pol}(\pi_0)$ where the entries of π_0 are i.i.d. generated from the uniform distribution in $(0, 1)$.

	GPB	Ad-GPB*		Pol-Subgrad	Pol-Ad-GPB*
$I1$	751s	600s		1390s	98s
$I2$	721s	550s		1503s	32s
$I3$	1250s	963s		5206s	98s

Table 5: Numerical results for solving LD subproblems. A relative tolerance of $\bar{\varepsilon} = 10^{-6}$ is set.

Table 5 shows that Ad-GPB* consistently outperforms GPB. Additionally, it shows that Pol-Ad-GPB* once again surpasses all other methods.

6 Concluding remarks

This paper presents a parameter-free adaptive proximal bundle method featuring two key ingredients: i) an adaptive strategy for selecting variable proximal step sizes tailored to specific problem instances, and ii) an adaptive cycle-stopping criterion that enhances the effectiveness of serious steps. Computational experiments reveal that our method significantly reduces the number of consecutive null steps (i.e., shorter cycles) while maintaining a manageable number of serious steps. As a result, it requires fewer iterations than methods employing a constant proximal step size and a non-adaptive cycle termination criterion. Moreover, our approach demonstrates considerable robustness to variations in the initial step size provided by the user.

We now discuss some possible extensions of our results. Recall that the complexity analysis of Ad-GPB* assumes that $\operatorname{dom} h$ is bounded, i.e., Lemma 4.2(c) uses it to give a simple proof that t_{i_k} is bounded in terms of (M, L) and the diameter D of $\operatorname{dom} h$. Using more complex arguments as those used in Appendix A of [19], we can show that the complexity analysis of Ad-GPB* can be extended to the setting where $\operatorname{dom} h$ is unbounded.

We finally discuss some possible extensions of our analysis in this paper. First, establishing the iteration complexity for Ad-GPB to the case where ϕ_* is unknown and $\operatorname{dom} h$ is unbounded is a more challenging and interesting research topic. Second, it would be interesting to analyze the complexities of Pol-GPB and Pol-Ad-GPB* described in Subsection 5 as they both performed well in our computational experiments. Finally, our current analysis does not apply to the one-cut bundle update scheme (see Subsection 3.1 of [19]) since it is not a special case of BUF as already observed in the second remark following BUF. It would be interesting to extend the analysis of this paper to establish the complexity of Ad-GPB based on the one-cut bundle update scheme.

References

[1] G. Angulo, S. Ahmed, and S. S Dey. Improving the integer l-shaped method. *INFORMS Journal on Computing*, 28(3):483–499, 2016.

[2] J. F. Bonnans, C. Lemaréchal J. C. Gilbert, and C. A. Sagastizábal. A family of variable metric proximal methods. *Mathematical Programming*, 68:15–47, 1995.

[3] Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 24(4):1779–1814, 2014.

[4] W. de Oliveira and M. Solodov. A doubly stabilized bundle method for nonsmooth convex optimization. *Mathematical programming*, 156(1-2):125–159, 2016.

[5] M. Díaz and B. Grimmer. Optimal convergence rates for the proximal bundle method. *SIAM Journal on Optimization*, 33(2):424–454, 2023.

[6] Y. Du and A. Ruszczyński. Rate of convergence of the bundle method. *Journal of Optimization Theory and Applications*, 173:908–922, 2017.

[7] A. M. Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10:237–260, 1972.

[8] V. Guigues, J. Liang, and R.D.C Monteiro. Universal subgradient and proximal bundle methods for convex and strongly convex hybrid composite optimization. *arXiv preprint arXiv:2407.10073*, 2024.

[9] Y. He and R. D. C. Monteiro. Accelerating block-decomposition first-order methods for solving composite saddle-point and two-player nash equilibrium problems. *SIAM Journal on Optimization*, 25(4):2182–2211, 2015.

[10] Y. He and R. D. C. Monteiro. An accelerated hpe-type algorithm for a class of composite convex-concave saddle-point problems. *SIAM Journal on Optimization*, 26(1):29–56, 2016.

[11] N. Karmitsa and M. M. Mäkelä. Adaptive limited memory bundle method for bound constrained large-scale nonsmooth optimization. *Optimization*, 59(6):945–962, 2010.

[12] K. C Kiwiel. Efficiency of proximal bundle methods. *Journal of Optimization Theory and Applications*, 104(3):589, 2000.

[13] K. C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, 16(4):1007–1023, 2006.

[14] O. Kolossoski and R. D. C. Monteiro. An accelerated non-euclidean hybrid proximal extragradient-type algorithm for convex–concave saddle-point problems. *Optimization Methods and Software*, 32(6):1244–1272, 2017.

[15] C. Lemaréchal. An extension of davidon methods to non differentiable problems. In *Nondifferentiable optimization*, pages 95–109. Springer, 1975.

[16] C. Lemaréchal. Nonsmooth optimization and descent methods. 1978.

[17] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: from conceptual to implementable forms. *Mathematical Programming*, 76:393–410, 1997.

[18] J. Liang and R. D. C. Monteiro. A proximal bundle variant with optimal iteration-complexity for a large range of prox stepsizes. *SIAM Journal on Optimization*, 31(4):2955–2986, 2021.

[19] J. Liang and R. D. C. Monteiro. A unified analysis of a class of proximal bundle methods for solving hybrid convex composite optimization problems. *Mathematics of Operations Research*, 49(2):832–855, 2024.

[20] J. Liang, R. D. C. Monteiro, and H. Zhang. Proximal bundle methods for hybrid weakly convex composite optimization problems. *arXiv preprint arXiv:2303.14896*, 2023.

[21] R. Mifflin. *A modification and an extension of Lemarechal's algorithm for nonsmooth minimization*, pages 77–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982.

[22] R. D. C. Monteiro and B. F. Svaiter. Complexity of variants of Tseng's modified F-B splitting and korpelevich's methods for hemivariational inequalities with applications to saddle-point and convex optimization problems. *SIAM Journal on Optimization*, 21(4):1688–1720, 2011.

[23] A. Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.*, 15(1):229–251, 2004.

[24] A. S. Nemirovski and D. B. Yudin. Cesari convergence of the gradient method of approximating saddle points of convex-concave functions. In *Doklady Akademii Nauk*, volume 239, pages 1056–1059. Russian Academy of Sciences, 1978.

[25] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Programming*, 103(1):127–152, 2005.

[26] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[27] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. In *Nondifferentiable optimization*, pages 145–173. Springer, 1975.

[28] J. Zou, S. Ahmed, and X. A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175:461–502, 2019.