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

Renato D.C. Monteiro * Honghao Zhang *

October 27, 2024 (first revision: August 24, 2025, second revision: April 28, 2026)

Abstract

This paper develops a parameter-free and line-search-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 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 two parameter-free and line-search-free adaptive proximal bundle methods for solving the HCCO problem (1).

Literature review on proximal bundle (PB) methods: 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. [6, 7, 14, 18, 19, 25, 31]) perform the serious step when x_j satisfies a relaxed

*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 Grants FA9550-22-1-0088 and FA9550-25-1-0182.

descent condition (e.g., see the paragraph containing equation (15) in [22]), which in its unrelaxed form implies that $\phi(x_j) \leq \phi(x^c)$. On the other hand, a new class of PB methods (see e.g. [9, 22, 23, 24]) perform the 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 [22, 23] that if $\lambda > 0$ is such that $\max\{\lambda, \lambda^{-1}\} = \mathcal{O}(\varepsilon^{-1})$, then the new class of 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 [6, 14] show that 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.

Despite their strong theoretical guarantees, the practical performance of the proximal bundle methods is highly sensitive to the choice of stepsize. Large proximal stepsizes can lead to long cycles, making the development of adaptive strategies to update the stepsizes particularly important. A natural strategy, upon observing a long cycle, is to reduce the proximal stepsize at the end of it before initiating a new one. However, a key drawback of this approach is its delay corrective action - it takes place at the end of a (possibly long) cycle.

Several papers of the classical PB methods (see e.g. [3, 5, 6, 12, 15, 20] of which only [6] deals with complexity analysis) have proposed ways of generating variable (but, nonadaptive) prox stepsizes to improve their computational performance. More recently, [9] developed a new variant of 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 [9] 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.

Main Contribution. The goal of this paper is to develop parameter-free and line-search-free adaptive PB methods, namely Ad-GPB* and Ad-GPB, with two important features: 1) adaptive choice of variable prox stepsizes that "closely fit" the instance under consideration; and 2) an adaptive criterion for making the occurrence of serious steps easier. The next two paragraphs further elaborate on these two features.

A key observation underlying the PB methods is that the cycle length can be controlled by ensuring that a key inequality holds at every null iteration. Motivated by this observation, our method employs the adaptive stepsize strategy which halves the stepsize after a null iteration that violates the key inequality and immediately proceeds to the next one. Because this inequality is parameter-free, the resulting procedure is line-search-free and does not require any knowledge of the parameters (M, L). More importantly, it generates cycles that are substantially shorter than those produced by its closest counterparts in [22, 23].

To motivate the relaxed cycle termination rule mentioned in 2), we briefly compare the theoretical cycle-length behavior of the classical [6, 7, 14, 18, 19, 25, 31] and the new [9, 22, 23] PB methods. In theory, classical PB methods perform on average $\mathcal{O}(\varepsilon^{-2})$ consecutive null iterations while the new PB methods perform only $\mathcal{O}(\varepsilon^{-1})$ consecutive null iterations in the worst case. This improvement stems from the more permissive δ-criterion used by the new class of PB methods to terminate 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 new class of PB methods. More specifically, under the simplifying assumption that ϕ_* is known, an Ad-GPB* cycle stops when the inequality $\phi(y_j) - m_j \leq \delta + [\phi(y_j) - \phi_*]/2$ is satisfied. The addition of the (usually large) term $[\phi(y_j) - \phi_*]/2$ makes this inequality easier to satisfy, thereby resulting in Ad-GPB* performing shorter cycles. Even though the previous observation assumes that ϕ_* is known, the Ad-GPB variant removes this assumption by instead using a lower bound on ϕ_* in the above inequality, obtained at the expense of assuming that the domain of h is bounded.

Finally, 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 fewer iterations than the GPB method of [9, 22, 23]. Moreover, in contrast to GPB, Ad-GPB* is robust to initial stepsize choices. In addition, we use Ad-GPB* to solve a relevant class of Lagrangian cut subproblems that arise while solving binary stochastic integer programming problems.

Other related methods: Another method related to, and subsequently developed from, proximal bundle methods is the bundle-level method, which was first proposed in [21] and later extended in various ways

(see, e.g., [2, 13, 17]). Finally, paper [5] presents a doubly stabilized bundle method whose prox subproblems combine elements from both proximal bundle and bundle-level methods.

Organization of the paper. §1.1 presents basic definitions and notation used throughout the paper. §2 contains two subsections. §2.1 formally describes problem (1) and the assumptions made on it. §2.2 reviews the GPB method and presents the motivation of this paper. §3 contains three subsections. §3.1 presents the Ad-GPB* framework and states the main iteration-complexity result for Ad-GPB*. §3.2 presents two special bundle update schemes. §3.3 contains two subsections. §3.3.1 provides a bound on the number of iterations within a cycle of Ad-GPB*. §3.3.2 establishes bounds on the number of cycles and the total number of iterations performed by Ad-GPB*. §4 presents the Ad-GPB framework under the assumption that ϕ_* is unknown and states the main iteration-complexity result for Ad-GPB, along with its proof. §5 presents the numerical results comparing Ad-GPB* and Ad-GPB with the two-cut bundle update scheme against other bundle methods. Finally, §6 presnets the concluding remarks.

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 $\log_2(\cdot)$ denote the logarithm to base 2, and define $\log_2^+(\cdot) := \max\{\log_2(\cdot), 0\}$. Similarly, let $\ln(\cdot)$ denote the natural logarithm, and define $\ln^+(\cdot) := \max\{\ln(\cdot), 0\}$. 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 $\overline{\operatorname{Conv}}(\mathbb{R}^n)$. $\mathbf{1}$ denotes the all one vector in $\mathbb{R}^n$.

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.

2 Main problem and motivation

This section contains two subsections. The first one describes the main problem and corresponding assumptions. The second one presents the motivation Ad-GPB*.

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 some 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, Assumption (A3) is used to model heterogeneous convex problems that contain both nonsmooth and smooth components. Indeed, when $L = 0$, (A3) reduces to a bounded-subgradient-type condition commonly used in nonsmooth convex optimization, while when $M = 0$, it reduces to the standard Lipschitz-gradient condition for smooth convex optimization. Similar hybrid conditions have been used in the study of hybrid convex composite optimization and related bundle-type methods (see, e.g., [22, 23]). Fourth, 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 $x, y \in \operatorname{dom} h$,

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

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

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

Finally, 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.,

$$d_0 := \|\hat{x}_0 - x^*\|, \quad \text{where} \quad x^* := \operatorname{argmin}\{\|\hat{x}_0 - x^*\| : x^* \in X^*\}. \tag{9}$$

2.2 Motivation for this work

The main purpose of this subsection is to motivate the variable-stepsize proximal bundle methods presented in this paper in light of the constant stepsize proximal bundle method of [23].

We start by outlining the PB method of [23], referred to as GPB (as in [23]) from now on.

GPB

Input: $(x_0, \varepsilon) \in \mathbb{R}^n \times \mathbb{R}^{++}$, $\lambda \in \mathbb{R}^{++}$

0. find $\Gamma_1 \in \mathcal{B}(\phi)$ such that $\Gamma_1 \geq \ell_\phi(\cdot; x_0)$ and set $y_0 = x^c = x_0$, and $j = 1$;
1. compute the optimal solution x_j and optimal value m_j of

$$\operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \Gamma_j(u) + \frac{1}{2\lambda}\|u - x^c\|^2 \right\}, \tag{10}$$

   and set $y_j := \operatorname{argmin}\{\phi(x) : x \in \{y_{j-1}, x_j\}\}$ and $t_j := \phi(y_j) - m_j$;
2. a) **if** $t_j \leq \varepsilon/2$ is violated **then** use x_j to update Γ_j to a new bundle $\Gamma_{j+1} \in \mathcal{B}(\phi)$;
   b) **else** compute $\Gamma_{j+1} \in \mathcal{B}(\phi)$ such that $\Gamma_{j+1} \geq \ell_\phi(\cdot; x_j)$ and set $x^c \leftarrow x_j$;
3. set $j \leftarrow j + 1$ and go to step 1.

We now provide several remarks on GPB method. First, an iteration j is referred to as a *null iteration* if step 2.a) is executed; otherwise, it is called a *serious iteration*. Second, a *cycle* is defined as a sequence of consecutive null iterations occurring between two consecutive serious iterations. Third, even though the above GPB outline does not specify how the point x_j is used to update the bundle Γ_j in step 2.a), the details of this update procedure are discussed in §3.2. Fourth, it is shown in [23] that $t_j - \varepsilon/4$ converges geometrically. Specifically, it is proven that if j is a null iteration, then $t_j - \varepsilon/4 \leq \tau_\lambda \left(t_{j-1} - \varepsilon/4\right)$, where τ_λ is the unique positive scalar satisfying

$$\frac{\tau_\lambda}{1-\tau_\lambda} = \frac{8\lambda(M^2 + L\varepsilon)}{\varepsilon}. \tag{11}$$

As a result, it is shown that the number of consecutive null iterations is $\tilde{\mathcal{O}}\left(1/(1-\tau_\lambda)\right)$. If the prox stepsize λ of GPB is large, and hence τ_λ is close to 1, the geometric convergence becomes slow, which may cause GPB to generate long cycles, and hence perform poorly in practice.

Motivation for our work: Our goal is to develop a parameter and line-search free adaptive variant of GPB method, called Ad-GPB, which chooses the prox stepsize more efficiently and generates shorter cycles than GPB. Its two main ideas are: i) its prox stepsizes are allowed to change within a cycle and each inner iteration adaptively chooses the prox stepsize for the next iteration based on a key test inequality so that no backtracking and line search is performed; and ii) its cycle stopping criterion is relaxed, thereby resulting in short cycles while still ensuring convergence of Ad-GPB. Below, we elaborate on each of these two ideas.

i) A natural strategy for avoiding long cycles is to choose λ such that $\tau_\lambda \leq \tau$ for some fixed $\tau \in (0,1)$ not too close to one (i.e., $\tau = 0.95$). However, this strategy requires prior knowledge of the parameters M and L, which are generally not known. We now discuss two parameter-free strategies for prox stepsize selection. These strategies are motivated by the fact that the condition $\tau_\lambda \leq \tau$ implies that

$$t_j - \frac{\varepsilon}{4} \leq \tau \left(t_{j-1} - \frac{\varepsilon}{4}\right) \tag{12}$$

for every null iteration j. Both strategies allow λ to change within a cycle in an attempt to either make it satisfy or become closer to satisfy (12). Specifically, in a null iteration j, both strategies compute x_j and m_j using (10) with λ replaced by λ_j, and then (y_j, t_j) as in step 1 of GPB. Hence, verification of (12) for some λ_j requires the computation of the exact solution of a subproblem as in (10).

The first and most obvious strategy is to perform a line search that halves λ_j and stops when (12) is satisfied. However, this strategy has the two drawbacks: i) it throws away the λ_j's, and hence the corresponding x_j's, that violate (12); and, ii) it is not forgiving in that it aggressively reduces the prox stepsize.

The second, more forgiving, strategy, which is the one studied in this paper, follows a *delayed update strategy*: given a pair (Γ_j, λ_j), the iterate x_j (and the next bundle Γ_{j+1}) is obtained by performing a GPB iteration with $\lambda = \lambda_j$ regardless of whether (12) is satisfied or not, and the next stepsize λ_{j+1} is set to λ_j if (12) is satisfied, or $\lambda_j/2$ otherwise. Thus, instead of enforcing (12) immediately at iteration j, the rationale of this strategy is to slowly "approach" (12). Its key advantage is that it *never backtracks* and is therefore line-search free.

ii) The cycle termination inequality in step 2.a) of GPB is relaxed by introducing an additional positive term to its right hand side. Specifically, if ϕ_* is known, a cycle of our new proposed variant of GPB stops when the inequality $\phi(y_j) - m_j \leq \delta + [\phi(y_j) - \phi_*]/2$ is satisfied. The addition of the (usually large) term $[\phi(y_j) - \phi_*]/2$ makes this inequality easier to satisfy, thereby resulting short cycles. When ϕ_* is not known, a strategy similar to the one above is developed where ϕ_* is replaced by a suitable lower bound.

3 Main Algorithm

This section contains three subsections. The first subsection presents Ad-GPB* that requires the optimal value ϕ_* of (1) and states its main complexity result. The second subsection presents two special bundle

update schemes. The third subsection provides the proof of the main complexity result.

Throughout this section, we assume that ϕ_* is known.

3.1 Main Algorithm

We first 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

$$\hat{x}_k \approx \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \phi(u) + \frac{1}{2\hat{\lambda}_k} \|u - \hat{x}_{k-1}\|^2 \right\}. \tag{13}$$

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

$$x_j = \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ \Gamma_j(u) + \frac{1}{2\lambda_j} \|u - \hat{x}_{k-1}\|^2 \right\}, \tag{14}$$

where Γ_j is a bundle approximation of ϕ and $\lambda_j \leq \lambda_{j-1}$ 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 (13). In contrast to the GPB method of [23], 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 gap for (1), a feature that considerably speeds up the computation of $\hat{x}_k$ for many subproblems (13).

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 §1.1 and the following notion of shadow function.

Definition 3.1 *Function $\bar{\Gamma}_j(\cdot) \in \mathcal{B}(\phi)$ is called a shadow of $\Gamma_j(\cdot)$ for (2) if it satisfies*

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

where x_j is the optimal solution of (2).

We are now ready to state the Ad-GPB* algorithm.

Ad-GPB*

Input: $(\hat{x}_0, \varepsilon) \in \mathbb{R}^n \times \mathbb{R}_{++}$, $(\hat{\lambda}_0, \tau) \in \mathbb{R}_{++} \times (0,1)$

0. set $y_0 = \hat{y}_0 = \hat{x}_0$, $j_0 = 0$, $j = k = 1$;

1. **(initializing the k-th cycle)** choose $\lambda_j \in [\hat{\lambda}_{k-1}, 2\hat{\lambda}_{k-1}]$, set $\Gamma_j = \ell_\phi(\cdot; \hat{x}_{k-1})$, and compute k-th cycle tolerance δ_k as
$$\delta_k = \frac{\phi(\hat{y}_{k-1}) - \phi_*}{4} + \frac{\varepsilon}{4}; \tag{15}$$

2. compute the optimal solution x_j and optimal value m_j of (14), pair (y_j, t_j) as
$$y_j := \operatorname{argmin} \{\phi(x) : x \in \{y_{j-1}, x_j\}\}, \quad t_j := \phi(y_j) - m_j; \tag{16}$$

3. **if** $t_j \leq \delta_k$ is violated **then** perform a **null update**, i.e.:

   3.a **(computing the next prox stepsize and bundle)**

∗ **if** either $j = j_{k-1} + 1$ or

$$t_j - \tau t_{j-1} \leq (1-\tau)\frac{\delta_k}{2}, \tag{17}$$

then set $\lambda_{j+1} = \lambda_j$; **else**, set $\lambda_{j+1} = \lambda_j/2$;

∗ let $\bar{\Gamma}_j$ be a shadow of Γ_j for (14) and set

$$\Gamma_{j+1}(\cdot) = \max\left\{\ell_\phi(\cdot; x_j), \bar{\Gamma}_j(\cdot)\right\}; \tag{18}$$

∗ set $j \leftarrow j+1$ and go to step 2;

else perform a **serious update**, i.e.:

3.b **(updating the next prox center $\hat{x}_k$)**

∗ set $\hat{x}_k = x_j$, $j_k = j$, and $(\hat{\lambda}_k, \hat{y}_k, \hat{\Gamma}_k, \hat{m}_k, \hat{t}_k) = (\lambda_j, y_j, \Gamma_j, m_j, t_j)$;

∗ **if** $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$, **then** output $\hat{y}_k$, and **stop**;

∗ set $k \leftarrow k+1$ and $j \leftarrow j+1$, 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 $j_1 \leq j_2 \leq \ldots$ denote the sequence of all serious iterations (i.e., the ones ending with a serious update) and, for every $k \geq 1$, define $i_k = j_{k-1} + 1$ and the k-th cycle $\mathcal{C}_k$ as

$$\mathcal{C}_k := \{i_k, \ldots, j_k\}. \tag{19}$$

Hence, index k counts the cycles generated by Ad-GPB*. Observe that steps 1 and 3.b of Ad-GPB* imply that for every $k \geq 1$,

$$\lambda_{i_k} \in [\hat{\lambda}_{k-1}, 2\hat{\lambda}_{k-1}], \quad \lambda_{j_k} = \hat{\lambda}_k. \tag{20}$$

An iteration $j \in \mathcal{C}_k$ is called good (resp., bad) if $\lambda_{j+1} \geq \lambda_j$ (resp., $\lambda_{j+1} = \lambda_j/2$). Observe that i_k and j_k are always good iterations in view of (20), the fact that $i_k = j_{k-1} + 1$ and the test in step 3.a of Ad-GPB*. Moreover, (17) is violated whenever $j \in \mathcal{C}_k \setminus \{j_k, i_k\}$ is a bad iteration.

We next make a remark about the quantities related to different Γ-functions that appear in Ad-GPB*. Step 1 of Ad-GPB*, the definitions of $\bar{\Gamma}_j$ and $\mathcal{B}(\phi)$ in Definition 3.1 and §1.1, respectively, and (18) imply that

$$\Gamma_j \leq \phi, \quad \Gamma_j \in \overline{\mathrm{Conv}}\,(\mathbb{R}^n), \quad \mathrm{dom}\,\Gamma_j = \mathrm{dom}\,h \quad \forall j \geq 1, \tag{21}$$

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 $\hat{\Gamma}_k \leq \phi$.

We now make some other relevant remarks about Ad-GPB*. First, it follows from (21) and the definition of x_j in (14) that $x_j \in \mathrm{dom}\,h$ for every $j \geq 1$. Second, an induction argument using (16) and the fact that $y_0 = \hat{x}_0 \in \mathrm{dom}\,h$ imply that $y_j \in \{\hat{x}_0, x_1, \ldots, x_j\} \subset \mathrm{dom}\,h$ and

$$y_j \in \mathrm{Argmin}\,\{\phi(x) : x \in \{\hat{x}_0, x_1, \ldots, x_j\}\} \tag{22}$$

(hence, $\phi(y_{j+1}) \leq \phi(y_j)$) for every $j \geq 1$. Third, if Ad-GPB* reaches step 1 then the primal gap $\phi(\hat{y}_{k-1}) - \phi_*$ is greater than ε because of its step 3.b and is substantially larger than this lower bound at its early cycles. Hence, the right-hand side of its cycle termination criterion δ_k in step 3 is always larger than $\varepsilon/2$ and is substantially larger than $\varepsilon/2$ 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 looser, and potentially much looser at its early cycles, than that of GPB. Thus Ad-GPB* allows earlier cycles to terminate in less number of inner iterations, and hence speeds up the overall performance of the method.

We now comment on the inexactness of $\hat{y}_k$ as a solution of prox subproblem (13) 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 (13) 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 3 holds, then we conclude that $\hat{y}_k$ is an ε_k-solution of (13), where

$$\varepsilon_k := \delta_k + \frac{\|\hat{y}_k - \hat{x}_{k-1}\|^2}{2\hat{\lambda}_k}.$$

Lastly, Ad-GPB* never restarts a cycle, i.e., attempts to inexactly solve two or more subproblems (13) 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 (13) with $\hat{\lambda}_k$ set to be the last λ_j generated within the k-th cycle (see step 3.a of Ad-GPB*).

We now state the main complexity result for Ad-GPB*.

Theorem 3.2 *Define*

$$\bar{t} := 4D(4M + LD), \qquad \bar{C} = \bar{C}(M, L, \hat{\lambda}_0, \tau, \varepsilon) := \frac{16(1-\tau)}{\tau}\left(16M^2 + L\varepsilon\right) + \frac{2\varepsilon}{\hat{\lambda}_0}, \tag{23}$$

$$\hat{K}(\varepsilon) := \left\lceil \left(2d_0^2 + 2\hat{\lambda}_0[\phi(\hat{x}_0) - \phi_*]\right)\frac{\bar{C}}{\varepsilon^2} \right\rceil, \tag{24}$$

where the tuple $(\hat{x}_0, \hat{\lambda}_0, \tau, \varepsilon)$ is the input to Ad-GPB. Then, Ad-GPB* finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$ in at most $\hat{K}(\varepsilon)$ cycles and*

$$\frac{3\hat{K}(\varepsilon)}{1-\tau}\left[\ln^+\left(\frac{\bar{t}}{\varepsilon}\right) + 1\right] + \log_2^+\left(\frac{\hat{\lambda}_0\bar{C}}{2\varepsilon}\right) \tag{25}$$

iterations.

We now make some remarks about Theorem 3.2. First, the iteration complexity of Ad-GPB* to find a ε-solution of (1) is

$$\tilde{\mathcal{O}}\left(\frac{\left(d_0^2 + \hat{\lambda}_0[\phi(\hat{x}_0) - \phi_*]\right)}{\varepsilon^2}\left(\frac{M^2 + \varepsilon L}{\tau} + \frac{\varepsilon}{(1-\tau)\hat{\lambda}_0}\right)\right).$$

Second, if an upper upper D on d_0 is known and $\hat{\lambda}_0$ satisfies

$$\hat{\lambda}_0 = \Theta\left(\frac{D^2}{\phi(\hat{x}_0) - \phi_*}\right),$$

then the inner-iteration complexity of Ad-GPB* becomes

$$\tilde{\mathcal{O}}\left(\frac{D^2}{\varepsilon^2}\left(\frac{M^2 + \varepsilon L}{\tau} + \frac{(\phi(\hat{x}_0) - \phi_*)\varepsilon}{(1-\tau)D^2}\right)\right). \tag{26}$$

Third, another way of choosing $\hat{\lambda}_0$ is a multiple of the Polyak stepsize, i.e.,

$$\hat{\lambda}_0 = \frac{\alpha(\phi(\hat{x}_0) - \phi_*)}{\|\phi'(\hat{x}_0)\|^2}$$

where the multiple factor α is such that $0 < \alpha = \mathcal{O}(1)$ and $\phi'(\hat{x}_0)$ is a subgradient of ϕ at $\hat{x}_0$. This choice of $\hat{\lambda}_0$ and (9) then imply that

$$\hat{\lambda}_0[\phi(\hat{x}_0) - \phi_*] = \frac{\alpha[\phi(\hat{x}_0) - \phi_*]^2}{\|\phi'(\hat{x}_0)\|^2} \leq \frac{\alpha\left(\langle\phi'(\hat{x}_0), \hat{x}_0 - x^*\rangle\right)^2}{\|\phi'(\hat{x}_0)\|^2} \leq \alpha\|\hat{x}_0 - x^*\|^2 \overset{(9)}{=} \alpha d_0^2.$$

where the first and second inequalities follow from the subgradient and the Cauchy–Schwarz inequalities, respectively. Thus, assuming without loss of generality that $\phi(\hat{x}_0) - \phi_* > \varepsilon$, the inner-iteration complexity bound (25) becomes

$$\tilde{\mathcal{O}}\left(\frac{d_0^2}{\varepsilon^2}\left(\frac{M^2+\varepsilon L}{\tau} + \frac{\|\phi'(\hat{x}_0)\|^2}{1-\tau}\right)\right). \tag{27}$$

Finally, we now comment on the dependence of (26) and (27) in terms of ε only. Indeed, if $\tau \in (0,1)$ satisfies $\tau^{-1} = \mathcal{O}(1)$ and $(1-\tau)^{-1} = \mathcal{O}(\varepsilon^{-1})$, then bound (26) is $\tilde{\mathcal{O}}(\varepsilon^{-2})$; moreover, if $\max\{\tau^{-1}, (1-\tau)^{-1}\} = \mathcal{O}(1)$, then bound (27) is $\tilde{\mathcal{O}}(\varepsilon^{-2})$.

3.2 Special instances of shadow

Noting that the shadow $\bar{\Gamma}_j$ in step 3.a of Ad-GPB* is undetermined, Ad-GPB* has the flexibility to choose $\bar{\Gamma}_j$. Next, we present two specific schemes of constructing $\bar{\Gamma}_j$, and hence two concrete ways to implement step 3.a of Ad-GPB*.

- **2-cut:** Now we describe how this scheme generates shadow $\{\bar{\Gamma}_j\}$ within k-th cycle. Every shadow generated by this scheme is of the form $\bar{\Gamma}_j = A_j + h$ where A_j is an affine function satisfying $A_j \le f$. Note that the form of $\bar{\Gamma}_j$ above, the definition of $\ell_\phi(\cdot;\cdot)$ in (8), and identity (18), imply that

  $$\Gamma_{j+1}(\cdot) = \max\left\{\ell_\phi(\cdot;x_j), \bar{\Gamma}_j(\cdot)\right\} = \max\left\{\tilde{\ell}_f(\cdot;x_j), A_j(\cdot)\right\} + h(\cdot),$$

  and hence that (14) is equivalent to

  $$x_{j+1} = \operatorname*{argmin}_{u\in\mathbb{R}^n, t\in\mathbb{R}}\left\{t + h(u) + \frac{1}{2\lambda_j}\|u-\hat{x}_{k-1}\|^2 :\ \tilde{\ell}_f(u;x_j) - t \le 0,\ A_j(u) - t \le 0\right\}. \tag{28}$$

  The sequence $\{A_j\}$, and hence $\{\bar{\Gamma}_j := A_j + h\}$, over the k-th cycle is recursively generated as

  $$A_{j+1}(\cdot) = \begin{cases} \tilde{\ell}_f(\cdot;\hat{x}_{k-1}), & \text{if } j = j_{k-1}, \\ \theta_j A_j(\cdot) + (1-\theta_j)\tilde{\ell}_f(\cdot;x_j), & \text{if } j \in \mathcal{C}_k \setminus \{j_k\}, \end{cases} \tag{29}$$

  where $\theta_j \in [0,1]$ for $j \in \mathcal{C}_k \setminus \{j_k\}$ is the Lagrangian multiplier for the second constraint $A_j(u) - t \le 0$ of (28). Using the optimality conditions for (28), it can be shown that $\bar{\Gamma}_j$ is a shadow of Γ_j for (14) for every $j \ge 1$.

- **multi-cut:** This scheme sets $\bar{\Gamma}_j(\cdot) = \Gamma_j(\cdot)$. It is obvious that $\bar{\Gamma}_j$ is a shadow of Γ_j for (14).

We now make a remark about step 3.a of Ad-GPB*. Instead of following (18), [23] replaces (18) by the weaker inequality $\Gamma^+(\cdot) \ge \tau\bar{\Gamma}(\cdot) + (1-\tau)\ell_\phi(\cdot;x)$ for some $\tau \in (0,1)$ and, as a result, contains the one-cut bundle update scheme described in §3.1 of [23].

3.3 Proof of Theorem

This subsection is divided into two parts. The first one establishes a bound on the lengths of the cycles generated by Ad-GPB*. The second one establishes a bound on the number of cycles generated by Ad-GPB* and provides the proof of Theorem 3.2.

3.3.1 Bounding cycle lengths of Ad-GPB*

Recall from (19) 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 3.3 *For every $j \in \mathcal{C}_k \setminus \{i_k\}$, the following statements hold:*

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

$$\max\left\{\bar{\Gamma}_{j-1}(\cdot), \ell_\phi(\cdot; x_{j-1})\right\} = \Gamma_j(\cdot) \le \phi(\cdot), \tag{30}$$

$$\bar{\Gamma}_{j-1} \in \mathrm{Conv}(\mathbb{R}^n), \quad \bar{\Gamma}_{j-1}(x_{j-1}) = \Gamma_{j-1}(x_{j-1}), \tag{31}$$

$$x_{j-1} = \operatorname*{argmin}_{u\in\mathbb{R}^n} \left\{\bar{\Gamma}_{j-1}(u) + \frac{1}{2\lambda_{j-1}}\|u - \hat{x}_{k-1}\|^2\right\}; \tag{32}$$

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

$$\bar{\Gamma}_{j-1}(u) + \frac{1}{2\lambda_{j-1}}\|u - \hat{x}_{k-1}\|^2 \ge m_{j-1} + \frac{1}{2\lambda_{j-1}}\|u - x_{j-1}\|^2. \tag{33}$$

Proof: a) This statement immediately follows from (18), the definition of $\bar{\Gamma}_j$ in Definition 3.1, all with $j = j - 1$.

b) Using (32) 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, the second identity in (31), and the definition of m_j in step 2 of Ad-GPB*. ∎

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

Lemma 3.4 *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*

$$t_j - \tau t_{j-1} \le \frac{4M^2(1-\tau)^2\lambda_{j-1}}{\tau}. \tag{34}$$

Proof: a) Inequality (30) implies that for every $\tau' \in [0, 1]$, we have

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

The definition of m_j in step 2 of Ad-GPB*, the above inequality, and (33) with $u = x_j$, imply that

$$\begin{aligned}
m_j &\ge (1-\tau')\ell_\phi(x_j; x_{j-1}) + \tau'\bar{\Gamma}_{j-1}(x_j) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2 \\
&= (1-\tau')\left[\ell_\phi(x_j; x_{j-1}) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] + \tau'\left[\bar{\Gamma}_{j-1}(x_j) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] \\
&\overset{\lambda_j \le \lambda_{j-1}}{\ge} (1-\tau')\left[\ell_\phi(x_j; x_{j-1}) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] + \tau'\left[\bar{\Gamma}_{j-1}(x_j) + \frac{1}{2\lambda_{j-1}}\|x_j - \hat{x}_{k-1}\|^2\right] \\
&\overset{(33)}{\ge} (1-\tau')\left[\ell_\phi(x_j; x_{j-1}) + \frac{1}{2\lambda_j}\|x_j - \hat{x}_{k-1}\|^2\right] + \tau'\left[m_{j-1} + \frac{1}{2\lambda_{j-1}}\|x_j - x_{j-1}\|^2\right].
\end{aligned}$$

Using this inequality and the definition of t_j in (16), 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 (16). 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 $j \in \mathcal{C}_k \setminus \{i_k, j_k\}$ is a bad iteration of Ad-GPB*, then (17) is violated (see step 3.a of Ad-GPB* and the first paragraph following Ad-GPB*).

Lemma 3.5 *Define*

$$\underline{\lambda}(\varepsilon) := \min\left\{\frac{\tau\varepsilon}{128(1-\tau)M^2}, \frac{\tau}{8(1-\tau)L}\right\}, \tag{35}$$

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

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

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

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

$$\lambda_j < \min\left\{\underline{\lambda}(\varepsilon), \lambda_{i_k}\right\}, \tag{36}$$

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

$$\frac{\lambda_{j-2}}{4} \leq \frac{\lambda_{j-1}}{2} = \lambda_j. \tag{37}$$

Before showing the claim, we argue that (37) implies the conclusion of the lemma. Indeed, noting that (35) and (37) implies that $\lambda_{j-2} \leq 4\underline{\lambda}(\varepsilon) \leq \tau/(2(1-\tau)L)$, it follows from (34) with $j = j-1$ and the definition of $\underline{\lambda}(\varepsilon)$ in (35) that

$$\begin{aligned}
t_{j-1} - \tau t_{j-2} &\overset{(34)}{\leq} \frac{4(1-\tau)^2\lambda_{j-2}M^2}{\tau} \leq \frac{16(1-\tau)^2\lambda_j M^2}{\tau} \leq \frac{16(1-\tau)^2\underline{\lambda}(\varepsilon)M^2}{\tau} \\
&\overset{(35)}{\leq} (1-\tau)\frac{\varepsilon}{8} \leq (1-\tau)\frac{\delta_k}{2},
\end{aligned}$$

where the last equality is due to the fact that $\delta_k \geq \varepsilon/4$ in view of the definition of δ_k in (15). This conclusion then implies that (17) holds for iteration $j-1$, and hence that $\lambda_j = \lambda_{j-1}$ due to the logic of step 3.a of Ad-GPB*. Since this contradicts (37), statement (a) follows.

We will now show the above claim, i.e., that the definition of j implies (37). Indeed, since the logic of step 3.a implies that $\lambda_{i_k+1} = \lambda_{i_k}$ and j is the smallest index in $\mathcal{C}_k$ satisfying (36), 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 3.a 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 (37) hold.

b) First observe that for any $j \in \{i_k, \ldots, j_k - 1\}$, we have $\lambda_{j+1} = \lambda_j$ (resp., $\lambda_{j+1} = \lambda_j/2$) if j is a good (resp., bad) iteration. Using this observation, it then follows that $\lambda_{i_k}/\hat{\lambda}_k = 2^{s_k}$. This observation together with (a) then implies that statement (b) holds. ∎

We now make some remarks about Lemma 3.5. First, (35) describes how the lower threshold $\bar{\lambda}$ on the adaptive stepsize implicitly captures the extreme cases of (M, L). Indeed, $\underline{\lambda}(\varepsilon) = \Omega(\varepsilon/M^2)$ in the nonsmooth case where $L = 0$ while $\underline{\lambda}(\varepsilon) = \Omega(1/L)$ in the smooth case where $M = 0$. Thus, the adaptive stepsize automatically adjusts to the problem structure without requiring any knowledge of (M, L).

Second, it follows from Lemma 3.5(b) that the number of bad iterations within the k-th cycle $\mathcal{C}_k$ is finite. Proposition 3.8 below provides a bound on $|\mathcal{C}_k|$, and hence shows that every cycle $\mathcal{C}_k$ terminates. Before showing this result, we state two technical results which provides some key properties about the sequence $\{t_j\}$.

Lemma 3.6 *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

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

Proof: a) The statement immediately follows from Lemma 3.4(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 (17) is satisfied and the cycle-stopping criterion is violated at iteration j, i.e.,

$$t_j > \delta_k. \tag{38}$$

These two observations and the fact that $\tau < 1$ then imply that

$$\begin{aligned} t_j - \frac{\varepsilon}{8} &\overset{(17)}{\leq} \tau t_{j-1} + (1-\tau)\frac{\delta_k}{2} - \frac{\varepsilon}{8} \\ &\overset{(38)}{\leq} \tau t_{j-1} + (1-\tau)\frac{t_j}{2} - \frac{\varepsilon}{8} \\ &\overset{\tau<1}{\leq} \tau\left(t_{j-1} - \frac{\varepsilon}{8}\right) + \frac{1-\tau}{2}\left(t_j - \frac{\varepsilon}{8}\right), \end{aligned}$$

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

∎

Lemma 3.7 *For every cycle index $k \geq 1$ generated by Ad-GPB*, the following statements hold:*

a) $t_{i_k} \leq \bar{t}/8$ where $\bar{t}$ is as in (23);

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

$$t_j - \frac{\varepsilon}{8} \leq \left(\frac{2\tau}{1+\tau}\right)^{j-i_k-s_k}\left(t_{i_k} - \frac{\varepsilon}{8}\right) \tag{39}$$

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

Proof: a) Using the facts that $\phi = f + h$ and $\Gamma_{i_k}(\cdot) = \ell_\phi(\cdot; \hat{x}_{k-1})$ (see step 1 of Ad-GPB*), and the definition of t_j, m_j and y_j in step 2 of Ad-GPB*, respectively, we have

$$\begin{aligned} t_{i_k} &\overset{(16)}{=} \phi(y_{i_k}) - m_{i_k} \overset{(16)}{\leq} \phi(x_{i_k}) - m_{i_k} = \phi(x_{i_k}) - \Gamma_{i_k}(x_{i_k}) - \frac{1}{2\lambda_{i_k}}\|x_{i_k} - \hat{x}_{k-1}\|^2 \\ &= f(x_{i_k}) - \tilde{\ell}_f(x_{i_k}; \hat{x}_{k-1}) - \frac{1}{2\lambda_{i_k}}\|x_{i_k} - \hat{x}_{k-1}\|^2 \\ &\overset{(7)}{\leq} 2M\|x_{i_k} - \hat{x}_{k-1}\| + \frac{L}{2}\|x_{i_k} - \hat{x}_{k-1}\|^2 \\ &\leq 2MD + \frac{L}{2}D^2. \end{aligned} \tag{40}$$

where the last inequality is due to Assumption 1, and the fact that both x_{i_k} and $\hat{x}_{k-1}$ are in $\operatorname{dom} h$. Statement a) now follows from the above inequality and the definition of $\bar{t}$ in (23).

b) If $j - i_k - s_k \leq 0$, then (39) 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) of Lemma 3.6, imply that

$$t_j - \frac{\varepsilon}{8} \leq \left(t_{i_k} - \frac{\varepsilon}{8}\right)\left(\frac{2\tau}{1+\tau}\right)^{j - i_k - s_k} \tag{41}$$

and thus the statement follows from the above inequality. ∎

Proposition 3.8 *For every cycle index $k \geq 1$ generated by Ad-GPB*, the size of the k-th cycle is bounded by $|\mathcal{C}_k| \leq s_k + \bar{N}(\varepsilon) + 1$ where s_k denotes the number of bad iterations within it and $\bar{N}(\cdot)$ is defined as*

$$\bar{N}(\varepsilon) := \left\lceil \frac{2}{1-\tau} \ln^+\left(\frac{\bar{t}}{\varepsilon}\right)\right\rceil. \tag{42}$$

Proof: It suffices to prove that

$$|\mathcal{C}_k| \leq s_k + \bar{N}_k(\varepsilon) + 1$$

where

$$\bar{N}_k(\varepsilon) := \left\lceil \frac{\tau+1}{1-\tau} \ln^+\left(\frac{t_{i_k}}{\varepsilon}\right)\right\rceil. \tag{43}$$

Indeed, if the above claim is true, then the statement obviously follows from Lemma 3.7 (a), the definition of $\bar{N}(\varepsilon)$ in (42), and the fact that $\tau \leq 1$. Now we start to prove the above claim. 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

$$J - i_k + 2 > s_k + \bar{N}_k(\varepsilon) + 1 \tag{44}$$

because the left-hand side of (44) 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 3 of Ad-GPB is violated with $j = J$, i.e.,

$$t_J > \delta_k \geq \frac{\varepsilon}{4}$$

where the last inequality is due to the definition of δ_k in (15). This observation together with Lemma 3.6(c) with $j = J$ and statement a), 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 (43) can be easily seen to imply that

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

Since this conclusion contradicts (44), the claim is proved. ∎

Before ending §3.3.1, we now make an important remark about it. Even though the quantity δ_k defined in (15) appears in the proofs of Lemmas 3.5, 3.6, 3.7, and 3.8, the only fact that is used about it is that $\delta_k \geq \varepsilon/4$. Thus, all the results in the analysis of §3.3.1 also hold under the more general assumption that $\delta_k \geq \varepsilon/4$ for every $k \geq 1$.

3.3.2 Bounding number of cycles of Ad-GPB*

This subsection establishes a bound on the number of cycles generated by Ad-GPB*. Even though the expression of δ_k in (15) was not used in its full extent in the previous subsection, it will be so in this one.

We start by stating two technical results. The first result describes some basic facts about the sextuple $(\hat{\lambda}_k, \hat{x}_k, \hat{y}_k, \hat{\Gamma}_k, \hat{m}_k, \hat{t}_k)$ obtained in step 3.b of Ad-GPB*, and hence at the last iteration of its k-th cycle.

Lemma 3.9 *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) for every given $u \in \mathrm{dom}\,h$, *we have*

$$\phi(\hat{y}_k) - \hat{\Gamma}_k(u) \leq \hat{t}_k + \frac{1}{2\hat{\lambda}_k}\left[\|u - \hat{x}_{k-1}\|^2 - \|u - \hat{x}_k\|^2\right]; \tag{45}$$

Proof: a) This statement follows from (21) and the fact that $\hat{\Gamma}_k$ is the last Γ_j generated within the k-th cycle.

b) Observe that (21) and the definitions of the quantities $\hat{x}_k$, $\hat{m}_k$, $\hat{\Gamma}_k$, and $\hat{\lambda}_k$ in step 3.b of Ad-GPB*, imply that $(\hat{x}_k, \hat{m}_k)$ is the pair of optimal solution and optimal value of

$$\min\left\{\hat{\Gamma}_k(u) + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2 : u \in \mathbb{R}^n\right\}. \tag{46}$$

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

$$\hat{m}_k + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_k\|^2 \leq \hat{\Gamma}_k(u) + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2, \tag{47}$$

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 3.b of Ad-GPB*. This shows that (45), and hence statement (b), holds. ∎

The next result provides a uniform lower bound on the sequence $\hat{\lambda}_k$ and also a bound on the total number of bad iterations generated by Ad-GPB*.

Lemma 3.10 *The following statements hold for every cycle index $k \geq 1$ generated by Ad-GPB*:*

a) $\hat{\lambda}_k \geq \min\{\underline{\lambda}(\varepsilon), \hat{\lambda}_0\}$ *where* $\underline{\lambda}(\varepsilon)$ *is as in* (35);

b) $\sum_{l=1}^{k} s_l \leq \log_2^+(\hat{\lambda}_0/\underline{\lambda}(\varepsilon)) + k$ *where s_l denotes the number of bad iterations within cycle l.*

Proof: a) Using the facts that $\lambda_{i_k} \geq \hat{\lambda}_{k-1}$ in view of (20) and Lemma 3.5(a) with $j = j_k$, we conclude that

$$\hat{\lambda}_k \geq \min\left\{\underline{\lambda}(\varepsilon), \hat{\lambda}_{k-1}\right\}.$$

The statement then follows by using the above inequality recursively.

b) Recalling that s_l denotes the number of bad iterations in the l-th cycle. Observe that for any $j \in \{i_k, \ldots, j_k - 1\}$, we have λ_{j+1}/λ_j is equal to $1/2$ (resp., equal to 1) if j is a bad (resp., good) iteration. Using this observation, we easily see that $\hat{\lambda}_l/\lambda_{i_l} = (1/2)^{s_l}$. Using this inequality and the fact that $\lambda_{i_l} \leq 2\hat{\lambda}_{l-1}$ in view of (20), we can conclude $\log_2 \hat{\lambda}_{l-1} - \log_2 \hat{\lambda}_l + 1 \geq s_l$. The statement now follows by summing the last equality from $l = 1$ to k and using a). ∎

We now make two remarks about Lemma 3.10. First, it follows from Proposition 3.8 and Lemma 3.10(b) that the average number of inner iterations per outer performed by Ad-GPB* is $\tilde{\mathcal{O}}(1)$. Second, instead of the more relaxed condition (20), λ_{i_k} is chosen as $\lambda_{i_k} = \hat{\lambda}_{k-1}$, then it can actually be proved that the total (instead of the average) number of bad iterations is $\tilde{\mathcal{O}}(1)$.

The next lemma establishes an important primal gap bound that is used in the proof of Theorem 3.2.

Lemma 3.11 *If $k \geq 1$ is a cycle index generated by Ad-GPB*, then we have*

$$\sum_{l=1}^{k} \hat{\lambda}_l[\phi(\hat{y}_l) - \phi_*] \leq \hat{\lambda}_0[\phi(\hat{y}_0) - \phi_*] + \frac{\varepsilon}{2}\sum_{l=1}^{k} \hat{\lambda}_l + d_0^2. \tag{48}$$

Proof: Let $1 \leq l \leq k$ be fixed. It follows from the fact that the cycle-stopping criterion in the first line of step 3 of Ad-GPB* is satisfied at iteration j_k, the definition of δ_k in (15), and the definitions of the quantities $\hat{t}_k$ and $\hat{y}_k$ in step 3.b of Ad-GPB* that

$$\hat{t}_l \leq \frac{\phi(\hat{y}_{l-1}) - \phi_*}{4} + \frac{\varepsilon}{4}. \tag{49}$$

The above inequality, (45) with $u = x^*$, the definition of x^* in (9), and the facts that $\hat{\Gamma}_l \leq \phi$ and $\phi_* = \phi(x^*)$, imply that

$$\begin{aligned}
\hat{\lambda}_l[\phi(\hat{y}_l) - \phi_*] &\leq \hat{\lambda}_l[\phi(\hat{y}_l) - \hat{\Gamma}_l(x^*)] \\
&\leq \hat{\lambda}_l \hat{t}_l + \frac{1}{2}\|\hat{x}_{l-1} - x^*\|^2 - \frac{1}{2}\|\hat{x}_l - x^*\|^2 \\
&\overset{(49)}{\leq} \hat{\lambda}_l\left[\frac{\phi(\hat{y}_{l-1}) - \phi_*}{4} + \frac{\varepsilon}{4}\right] + \frac{1}{2}\|\hat{x}_{l-1} - x^*\|^2 - \frac{1}{2}\|\hat{x}_l - x^*\|^2 \\
&\leq 2\hat{\lambda}_{l-1}\left[\frac{\phi(\hat{y}_{l-1}) - \phi_*}{4}\right] + \frac{\varepsilon}{4}\hat{\lambda}_l + \frac{1}{2}\|\hat{x}_{l-1} - x^*\|^2 - \frac{1}{2}\|\hat{x}_l - x^*\|^2
\end{aligned}$$

where the last inequality is due to the fact that $\{\hat{\lambda}_k\}$ is non-increasing in view of the logic of Ad-GPB*. Summing the above inequality from $l = 1, \ldots, k$, we conclude that

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

The statement now follows from the above inequality and the definition of d_0 in (9). ∎

We are now ready to prove Theorem 3.2.

Proof: We first prove that Ad-GPB* finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \phi_* \leq \varepsilon$ in at most $\tilde{K}(\varepsilon)$ cycles where

$$\tilde{K}(\varepsilon) := \frac{2d_0^2 + 2\hat{\lambda}_0[\phi(\hat{y}_0) - \phi_*]}{\min\{\underline{\lambda}(\varepsilon), \hat{\lambda}_0\}\varepsilon}.$$

Suppose for contradiction that Ad-GPB* generates a cycle $K > \tilde{K}(\varepsilon)$. 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, the definition of $\tilde{K}(\varepsilon)$, inequality (48), the fact that $K-1 \geq \tilde{K}(\varepsilon)$, and Lemma 3.10(a), we conclude that

$$\frac{d_0^2+\hat{\lambda}_0[\phi(\hat{y}_0)-\phi_*]}{\varepsilon} \overset{(48)}{\geq} \frac{1}{\varepsilon}\sum_{k=1}^{K-1}\hat{\lambda}_k[\phi(\hat{y}_k)-\phi_*] - \frac{1}{2}\sum_{k=1}^{K-1}\hat{\lambda}_k > \sum_{k=1}^{K-1}\hat{\lambda}_k - \frac{1}{2}\sum_{k=1}^{K-1}\hat{\lambda}_k$$
$$\geq \frac{1}{2}\min\left\{\underline{\lambda}(\varepsilon),\hat{\lambda}_0\right\}(K-1) \geq \frac{1}{2}\min\left\{\underline{\lambda}(\varepsilon),\hat{\lambda}_0\right\}\tilde{K}(\varepsilon) = \frac{d_0^2+\hat{\lambda}_0[\phi(\hat{y}_0)-\phi_*]}{\varepsilon}$$

which yields the desired contradiction. Thus, the above claim is proved. Now the first statement of the theorem follows from the above claim and the fact that $\hat{K}(\varepsilon) \geq \tilde{K}(\varepsilon)$ in view of the definitions of $\hat{K}(\varepsilon)$ and $\underline{\lambda}(\varepsilon)$ in (24) and (35), respectively.

Let $\bar{k} \leq \hat{K}(\varepsilon)$ denote the numbers of cycles generated by Ad-GPB*. Proposition 3.8 and statement (b) of Lemma 3.10 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

$$\sum_{k=1}^{\bar{k}}|\mathcal{C}_k| \leq \sum_{k=1}^{\bar{k}}\left(\frac{2}{1-\tau}\ln^+\left(\frac{\bar{t}}{\varepsilon}\right)+2+s_k\right)$$
$$\leq \left(\frac{2}{1-\tau}\ln^+\left(\frac{\bar{t}}{\varepsilon}\right)+2\right)\hat{K}(\varepsilon)+\hat{K}(\varepsilon)+\log_2^+\frac{\hat{\lambda}_0}{\underline{\lambda}(\varepsilon)}$$

and hence by (25), due to the definitions of $\underline{\lambda}(\varepsilon)$ and $\bar{C}(M,L,\hat{\lambda}_0,\tau,\varepsilon)$ in (35) and (23), respectively, and the fact that $1 \leq 1/(1-\tau)$. ∎

We now make an important remark regarding §3. Even though we have assumed that $\operatorname{dom} h$ is bounded in §3, this assumption is only used in the proof of Proposition 3.7(a) to show that the quantity t_{i_k} is bounded. A more complex proof that t_{i_k} is bounded without assuming that $\operatorname{dom} h$ is bounded can be given using similar arguments to those used in the proof of Lemma 4.7 of [23]. Consequently, following the latter approach, the complexity analysis of Ad-GPB* can be extended to the setting where $\operatorname{dom} h$ is unbounded.

4 General case where ϕ_* is not known

This section presents a variant of Ad-GPB*, referred as Ad-GPB, for the case where ϕ_* is not known.

We start by formally describing Ad-GPB.

Ad-GPB

Input: $(\hat{x}_0,\varepsilon) \in \mathbb{R}^n \times \mathbb{R}_{++}$, $(\hat{\lambda}_0,\tau) \in \mathbb{R}^n \times (0,1)$

0. set $y_0=\hat{x}_0$, $j_0=0$, $\beta_1=1/4$, $j=k=1$, and
$$\hat{\ell}_0 = \inf_{u\in\operatorname{dom} h}\ell_\phi(u;\hat{x}_0); \tag{50}$$

1. **(initializing the k-th cycle)** choose $\lambda_j=\hat{\lambda}_{k-1}$, set $\Gamma_j=\ell_\phi(\cdot;\hat{x}_{k-1})$ and compute the k-th cycle tolerance δ_k as
$$\delta_k = \beta_k[\phi(\hat{y}_{k-1})-\hat{\ell}_{k-1}]+\frac{\varepsilon}{4}; \tag{51}$$

2. compute the optimal solution x_j and optimal value m_j of (14) and pair (y_j,t_j) as in (16);

3. **if** $t_j \leq \delta_k$ is violated **then** perform a **null update**, i.e.:

   3.a **(computing the next prox stepsize and bundle)**

   * **if** either $j=j_{k-1}+1$ or
   $$t_j - \tau t_{j-1} \leq (1-\tau)\frac{\delta_k}{2}, \tag{52}$$
   then set $\lambda_{j+1}=\lambda_j$; **else**, set $\lambda_{j+1}=\lambda_j/2$;

$*$ let $\bar{\Gamma}_j$ be a shadow of Γ_j for (14) and set

$$\Gamma_{j+1}(\cdot) = \max\left\{\ell_\phi(\cdot; x_j), \bar{\Gamma}_j(\cdot)\right\}; \tag{53}$$

$*$ set $j \leftarrow j+1$ and go to step 2;

else perform a **serious update**, i.e.:

3.b set $\hat{x}_k = x_j$, $j_k = j$, and $(\hat{\lambda}_k, \hat{y}_k, \hat{\Gamma}_k, \hat{m}_k, \hat{t}_k) = (\lambda_j, y_j, \Gamma_j, m_j, t_j)$;

3.c (**updating** $\hat{\ell}_k$) compute

$$\hat{\ell}_k := \max\left\{\hat{\ell}_{k-1}, \inf_{u\in\operatorname{dom} h} \mathcal{A}_k^a(u)\right\} \tag{54}$$

where

$$\mathcal{A}_k^a(u) := \frac{\sum_{l=\lceil k/2\rceil}^k \hat{\lambda}_l \mathcal{A}_l(u)}{\sum_{l=\lceil k/2\rceil}^k \hat{\lambda}_l}, \quad \mathcal{A}_k(u) := \hat{\Gamma}_k(\hat{x}_k) + \frac{1}{\hat{\lambda}_k}\langle \hat{x}_{k-1} - \hat{x}_k, u - \hat{x}_k\rangle; \tag{55}$$

3.d **if** $\phi(\hat{y}_k) - \hat{\ell}_k \le \varepsilon$, **then** output $(\hat{y}_k, \hat{\ell}_k)$, and **stop**;

3.e **(computing** β_{k+1}**)** compute

$$\hat{\Lambda}_k := \sum_{l=\lceil k/2\rceil}^k \hat{\lambda}_l, \quad \hat{\Phi}_k := \frac{\sum_{l=\lceil k/2\rceil}^k \hat{\lambda}_l \phi(\hat{y}_l)}{\hat{\Lambda}_k}, \quad \hat{g}_k := \frac{\sum_{l=\lceil k/2\rceil}^k \beta_l \hat{\lambda}_l \left[\hat{\ell}_k - \hat{\ell}_{l-1}\right]}{\hat{\Lambda}_k}; \tag{56}$$

if $\hat{g}_k \le (\hat{\Phi}_k - \hat{\ell}_k)/8$, **then** set $\beta_{k+1} = \beta_k$; **else**, set $\beta_{k+1} = \beta_k/2$;
set $k \leftarrow k+1$ and $j \leftarrow j+1$, and go to step 1.

end if

We now make some remarks about Ad-GPB.

First, the definition of $\mathcal{A}_k^a$ in (55) and relation (21) imply that $\mathcal{A}_k^a \in \overline{\operatorname{Conv}}\,(\mathbb{R}^n)$ and $\mathcal{A}_k^a \le \phi$, and hence that $\inf_u \mathcal{A}_k^a(u) \le \inf_u \phi(u) = \phi_*$.

Second, the facts that $\operatorname{dom}\mathcal{A}_k^a = \operatorname{dom} h$ is bounded (see Assumption (A1)) and $\mathcal{A}_k^a$ is a closed convex function imply that $\inf_u \mathcal{A}_k^a(u) > -\infty$. Moreover, the problem $\inf_u \mathcal{A}_k^a(u)$ 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 new one that is no worse, namely, $\hat{\ell}_k \ge \hat{\ell}_{k-1}$. For the sake of future reference, we note that

$$\phi_* \ge \hat{\ell}_k \ge \inf_{u\in\operatorname{dom} h} \mathcal{A}_k^a(u). \tag{57}$$

Third, if the test inequality in step 3.d holds, then the first component $\hat{y}_k$ of the pair output by Ad-GPB satisfies $\phi(\hat{y}_k) - \phi_* \le \varepsilon$ due to the fact that $\hat{\ell}_k \le \phi_*$.

Fourth, the cycle-stopping criterion, i.e., the inequality $t_j \le \delta_k$ in step 3, is a relaxation of the one used by GPB method of [23], in the sense its right-hand side δ_k has the extra term $\beta_k[\phi(\hat{y}_{k-1}) - \hat{\ell}_{k-1}]$ involving the relaxation factor β_k. 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. Moreover, the definition of δ_k in (51) is similar to that in (15), except that the constant 1/4 in (15) is replaced by β_k in (51) and ϕ_* in (15) is replaced by its lower bound $\hat{\ell}_k$ in (51). The quantities in (56) are used to update β_k at the end of the k-th cycle (see step 3.e of Ad-GPB).

Finally, assume that ϕ_* is known and $\hat{\ell}_0$ is chosen as $\hat{\ell}_0 = \phi_*$, instead of as in (50). Then, for every $k \ge 0$, (54) and the first inequality in (57) imply that $\hat{\ell}_k = \phi_*$, and hence that $\hat{g}_k = 0$, because of (56). Consequently, it follows from the test inequality in step 3.e of Ad-GPB that $\beta_{k+1} = \beta_k$ for all $k \ge 1$. Thus, Ad-GPB* with $\lambda_{i_k} = \hat{\lambda}_{k-1}$ can be viewed as the variant of Ad-GPB in which $\beta_1 = 1/4$ and $\hat{\ell}_0 = \phi_*$.

We now state the main complexity result for Ad-GPB.

Theorem 4.1 *Define*

$$\bar{K}(\varepsilon) := \left\lceil 8\left(D^2 + \hat{\lambda}_0(\phi(\hat{x}_0) - \hat{\ell}_0)\right)\frac{\bar{C}}{\varepsilon^2} + \log_2^+\left\{\frac{2(\phi_* - \hat{\ell}_0)}{\varepsilon}\right\} + 1\right\rceil \tag{58}$$

where $\bar{C} = \bar{C}(M, L, \hat{\lambda}_0, \tau, \varepsilon)$ is as in (23)*, the tuple $(\hat{x}_0, \hat{\lambda}_0, \tau, \varepsilon)$ is the input to Ad-GPB, and $\hat{\ell}_0$ is as in step 1 of Ad-GPB. Then, Ad-GPB finds a pair $(\hat{y}_k, \hat{\ell}_k) \in \operatorname{dom} h \times \mathbb{R}$ satisfying $\phi(\hat{y}_k) - \phi_* \leq \phi(\hat{y}_k) - \hat{\ell}_k \leq \varepsilon$ in at most $4\bar{K}(\varepsilon)$ cycles and*

$$8\bar{K}(\varepsilon)\left(\frac{1}{1-\tau}\ln^+\left(\frac{\bar{t}}{\varepsilon}\right) + 1\right) + \log_2^+\left(\frac{\hat{\lambda}_0\bar{C}}{2\varepsilon}\right) \tag{59}$$

iterations where $\bar{t}$ is as in (23)*.*

We now make some remarks about Theorem 4.1. First, the iteration complexity of Ad-GPB to find a ε-solution of (1) is

$$\tilde{\mathcal{O}}\left(\frac{\left(D^2 + \hat{\lambda}_0[\phi(\hat{x}_0) - \hat{\ell}_0]\right)}{\varepsilon^2}\left(\frac{M^2 + \varepsilon L}{\tau} + \frac{\varepsilon}{(1-\tau)\hat{\lambda}_0}\right)\right).$$

Second, if $\hat{\lambda}_0$ is chosen to satisfy

$$\hat{\lambda}_0 = \Theta\left(\frac{D^2}{\phi(\hat{x}_0) - \hat{\ell}_0}\right),$$

then the inner-iteration complexity of Ad-GPB becomes

$$\tilde{\mathcal{O}}\left(\frac{D^2}{\varepsilon^2}\left(\frac{M^2 + \varepsilon L}{\tau} + \frac{(\phi(\hat{x}_0) - \hat{\ell}_0)\varepsilon}{(1-\tau)D^2}\right)\right). \tag{60}$$

Third, another way of choosing $\hat{\lambda}_0$ is a multiple of a Polyak-type stepsize, i.e.,

$$\hat{\lambda}_0 = \frac{\alpha(\phi(\hat{x}_0) - \hat{\ell}_0)}{\|\phi'(\hat{x}_0)\|^2}$$

where the multiple factor α is such that $0 < \alpha = \mathcal{O}(1)$ and $\phi'(\hat{x}_0)$ is a subgradient of ϕ at $\hat{x}_0$. Noting that the definition of ℓ_ϕ in (8) and the fact that $\hat{\ell}_0 = \inf_{u\in\operatorname{dom} h}\ell_\phi(u; \hat{x}_0)$ imply that

$$\begin{aligned}
\phi(\hat{x}_0) - \hat{\ell}_0 &= \phi(\hat{x}_0) - \inf_{u\in\operatorname{dom} h}\left(f(\hat{x}_0) + \langle f'(\hat{x}_0), u - \hat{x}_0\rangle + h(u)\right) \\
&= h(\hat{x}_0) - \inf_{u\in\operatorname{dom} h}\left(\langle f'(\hat{x}_0), u - \hat{x}_0\rangle + h(u)\right) \\
&= \sup_{u\in\operatorname{dom} h}\left(\langle f'(\hat{x}_0), \hat{x}_0 - u\rangle + h(\hat{x}_0) - h(u)\right) \\
&\leq \sup_{u\in\operatorname{dom} h}\left(\langle f'(\hat{x}_0) + h'(\hat{x}_0), \hat{x}_0 - u\rangle\right) \leq \|\phi'(x_0)\|D
\end{aligned}$$

where the second last inequality follows from subgradient inequality and the last inequality follows from Cauchy–Schwarz inequality and Assumption A1. The above inequality and the choice of $\hat{\lambda}_0$ then imply that

$$\hat{\lambda}_0[\phi(\hat{x}_0) - \hat{\ell}_0] = \frac{\alpha[\phi(\hat{x}_0) - \hat{\ell}_0]^2}{\|\phi'(\hat{x}_0)\|^2} \leq \alpha D^2$$

Thus, assuming without loss of generality that $\phi(\hat{x}_0) - \hat{\ell}_0 > \varepsilon$, the inner-iteration complexity bound (59) becomes

$$\tilde{\mathcal{O}}\left(\frac{D^2}{\varepsilon^2}\left(\frac{M^2 + \varepsilon L}{\tau} + \frac{\|\phi'(\hat{x}_0)\|^2}{1-\tau}\right)\right). \tag{61}$$

Finally, we now comment on the dependence of (60) and (61) in terms of ε only. Indeed, if $\tau \in (0,1)$ satisfies $\tau^{-1} = \mathcal{O}(1)$ and $(1-\tau)^{-1} = \mathcal{O}(\varepsilon^{-1})$, then bound (60) is $\tilde{\mathcal{O}}(\varepsilon^{-2})$; moreover, if $\max\{\tau^{-1}, (1-\tau)^{-1}\} = \mathcal{O}(1)$, then bound (61) is $\tilde{\mathcal{O}}(\varepsilon^{-2})$.

The remaining of the section is devoted to proving Theorem 4.1. The next result describes properties of Ad-GPB that are similar to the ones derived for Ad-GPB*. Specifically, its statement a) is analogous to Proposition 3.8, while its statements b) and c) are analogous to statements a) and b) of Lemma 3.10.

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

a) the size of the k-th cycle is bounded by $|\mathcal{C}_k| \leq s_k + \bar{N}(\varepsilon) + 1$ where s_k denotes the number of bad iterations within it and $\bar{N}(\cdot)$ is defined as in (42);

b) $\hat{\lambda}_k \geq \min\{\underline{\lambda}(\varepsilon), \hat{\lambda}_0\}$ where $\underline{\lambda}(\varepsilon)$ is as in (35);

c) $\sum_{l=1}^{k} s_l \leq \log_2^+(\hat{\lambda}_0/\underline{\lambda}(\varepsilon))$ where s_l denotes the number of bad iterations within cycle l.

Proof: a) The proof of a) follows from Proposition 3.8 and the remark following it.

b) The arguments used for proving this statement are the same as those used in the proof of Lemma 3.10(a).

c) Recalling that s_l denotes the number of bad iterations in the l-th cycle. Observe that for any $j \in \{i_k, \ldots, j_k - 1\}$, we hvae λ_{j+1}/λ_j is equal to $1/2$ (resp., equal to 1) if j is a bad (resp., good) iteration. Using this observation and the fact that $\lambda_{i_k} = \hat{\lambda}_{k-1}$ in view of step 1 of Ad-GPB, we easily see that $\hat{\lambda}_l/\hat{\lambda}_{l-1} = (1/2)^{s_l}$, or equivalently, $\log_2 \hat{\lambda}_{l-1} - \log_2 \hat{\lambda}_l = s_l$. The statement now follows by summing the last equality from $l = 1$ to k and using b). ∎

It follows from Lemma 4.2(c) and the definition of $\underline{\lambda}(\varepsilon)$ in (35) that the overall number of bad iterations is

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

The next result describes some basic facts about the sextuple $(\hat{\lambda}_k, \hat{x}_k, \hat{y}_k, \hat{\Gamma}_k, \hat{m}_k, \hat{t}_k)$ obtained in step 3.b of Ad-GPB, and hence at the last iteration of its k-th cycle.

Lemma 4.3 *For every cycle index k of Ad-GPB, the following statements hold:*

a) we have

$$\hat{t}_k \leq \beta_k[\phi(\hat{y}_{k-1}) - \hat{\ell}_{k-1}] + \frac{\varepsilon}{4}; \tag{62}$$

b) $\hat{x}_k, \hat{y}_k \in \operatorname{dom} h$, $\phi_* \leq \phi(\hat{y}_k) \leq \hat{\Phi}_k$, and $\phi(\hat{y}_k) \leq \phi(\hat{y}_{k-1})$;

c) $\hat{\ell}_k \geq \hat{\ell}_{k-1}$, $\beta_{k+1} \leq \beta_k \leq 1/4$, and $\hat{\lambda}_{k+1} \leq \hat{\lambda}_k$;

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

$$\phi(\hat{y}_k) - \mathcal{A}_k(u) \leq \hat{t}_k + \frac{1}{2\hat{\lambda}_k}\left[\|u - \hat{x}_{k-1}\|^2 - \|u - \hat{x}_k\|^2\right]; \tag{63}$$

Proof: a) This statement follows from the fact that the cycle-stopping criterion in the first line of step 3 of Ad-GPB is satisfied at iteration j_k, the definition of δ_k in (51), and the definitions of the quantities $\hat{t}_k$ and $\hat{y}_k$ in step 3.b of Ad-GPB.

b) First, $\phi_* \leq \phi(\hat{y}_k)$ obviously holds. The rest of the statement follows from the fact that $x_j, y_j \in \operatorname{dom} h$ (see first two remarks in the paragraph containing (22)), the definition of $\hat{\Phi}_k$ in (56), 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.

c) The first inequality in (c) follows from the definition $\hat{\ell}_k$ in (54). Moreover, the update rule for β_{k+1} in Step 3.e of Ad-GPB ensures that $\beta_{k+1} \leq \beta_k$. Consequently, $\beta_k \leq \beta_1 = 1/4$, in view of step 0 of Ad-GPB. Finally, the update rule for λ_j in steps 1 and 3.a of Ad-GPB, together with the definition of $\hat{\lambda}_k$ in step 3.b, implies that $\hat{\lambda}_{k+1} \leq \hat{\lambda}_k$.

d) Observe that the definitions of the affine function $\mathcal{A}_k$ in (55) and the quantities $\hat{x}_k$, $\hat{m}_k$, $\hat{\Gamma}_k$, and $\hat{\lambda}_k$, in step 3.b of Ad-GPB, imply that $(\hat{x}_k, \hat{m}_k)$ is the pair of optimal solution and optimal value of

$$\min\left\{\mathcal{A}_k(u) + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2 : u \in \mathbb{R}^n\right\}. \tag{64}$$

The above observation and the fact that $\mathcal{A}_k(\cdot) + 1/(2\hat{\lambda}_k)\|\cdot - \hat{x}_{k-1}\|^2$ is $(1/\hat{\lambda}_k)$-strongly convex imply that for any $u \in \mathbb{R}^n$,

$$\hat{m}_k + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_k\|^2 \le \mathcal{A}_k(u) + \frac{1}{2\hat{\lambda}_k}\|u - \hat{x}_{k-1}\|^2. \tag{65}$$

The above relation together with the definition of $\hat{t}_k$ in step 3.b of Ad-GPB then imply that

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

for every $u \in \mathbb{R}^n$, and hence that statement (d) holds. ∎

The following lemma, which is analogous to Lemma 3.11, establishes a preliminary bound on $\hat{g}_k$ and the k-th dual gap $\hat{\Phi}_k - \hat{\ell}_k$ for Ad-GPB.

Lemma 4.4 *For every cycle index k of Ad-GPB, we have:*

$$\hat{\Phi}_k - \hat{\ell}_k \le \frac{\hat{\lambda}_0[\phi(\hat{x}_0) - \hat{\ell}_0]}{\hat{\Lambda}_k} + \frac{D^2}{\hat{\Lambda}_k} + \frac{\varepsilon}{2} + 2\hat{g}_k, \tag{66}$$

$$\hat{g}_k \le \beta_{\lceil \frac{k}{2} \rceil}\left(\phi_* - \hat{\ell}_0\right), \tag{67}$$

where D is as in (4) *and $\hat{\Phi}_k$ is as in* (56).

Proof: We first prove (66). Multiplying (63) by $\hat{\lambda}_l$ and using relation (62), it follows that for every $u \in \mathbb{R}^n$,

$$\begin{aligned}
2\hat{\lambda}_l[\phi(\hat{y}_l) - \mathcal{A}_l(u)] &\overset{(63)}{\le} 2\hat{\lambda}_l\hat{t}_l + \|\hat{x}_{l-1} - u_k^*\|^2 - \|\hat{x}_l - u\|^2 \\
&\overset{(62)}{\le} 2\hat{\lambda}_l\left[\beta_l\left(\phi(\hat{y}_{l-1}) - \hat{\ell}_{l-1}\right) + \frac{\varepsilon}{4}\right] + \|\hat{x}_{l-1} - u\|^2 - \|\hat{x}_l - u\|^2 \\
&= 2\hat{\lambda}_l\beta_l\left(\phi(\hat{y}_{l-1}) - \hat{\ell}_k\right) + 2\hat{\lambda}_l\beta_l\left(\hat{\ell}_k - \hat{\ell}_{l-1}\right) + \frac{\varepsilon}{2}\hat{\lambda}_l + \|\hat{x}_{l-1} - u\|^2 - \|\hat{x}_l - u\|^2 \\
&\le \hat{\lambda}_{l-1}\left(\phi(\hat{y}_{l-1}) - \hat{\ell}_k\right) + 2\hat{\lambda}_l\beta_l\left(\hat{\ell}_k - \hat{\ell}_{l-1}\right) + \frac{\varepsilon}{2}\hat{\lambda}_l + \|\hat{x}_{l-1} - u\|^2 - \|\hat{x}_l - u\|^2
\end{aligned} \tag{68}$$

where the last inequality uses the fact that (57) and Lemma 4.3(b) imply that $\phi(\hat{y}_{l-1}) - \hat{\ell}_k \ge 0$ and the inequalities $\beta_l \le 1/4$ and $\hat{\lambda}_{l+1} \le \hat{\lambda}_l$ for every $l \ge 1$ (see Lemma 4.3(c)). Let u_k^* be an optimal solution of $\min\{\mathcal{A}_k^a(u) : u \in \mathrm{cl}\,(\mathrm{dom}\, h)\}$, which exists in view of the assumption that $\mathrm{dom}\, h$ is bounded and the fact that $\mathcal{A}_k^a$ is a linear function. Summing the above inequality from $l = \lceil k/2 \rceil$ to $l = k$ with $u = u_k^*$, and using relation (57) and the definitions of $\mathcal{A}_k^a$ and $\hat{g}_k$ in (54) and (55), respectively, we have

$$\begin{aligned}
2\sum_{l=\lceil k/2 \rceil}^{k}\hat{\lambda}_l[\phi(\hat{y}_l) - \hat{\ell}_k] &\overset{(57)}{\le} 2\sum_{l=\lceil k/2 \rceil}^{k}\hat{\lambda}_l[\phi(\hat{y}_l) - \mathcal{A}_k^a(u_k^*)] \overset{(54)}{=} 2\sum_{l=\lceil k/2 \rceil}^{k}\hat{\lambda}_l[\phi(\hat{y}_l) - \mathcal{A}_l(u_k^*)] \\
&\overset{(68),(55)}{\le} \sum_{l=\lceil k/2 \rceil}^{k}\hat{\lambda}_{l-1}\left[\phi(\hat{y}_{l-1}) - \hat{\ell}_k\right] + 2\hat{\Lambda}_k\hat{g}_k + \frac{\hat{\Lambda}_k\varepsilon}{2} + \|\hat{x}_{\lceil k/2 \rceil - 1} - u_k^*\|^2 - \|\hat{x}_k - u_k^*\|^2 \\
&\le \sum_{l=\lceil k/2 \rceil}^{k}\hat{\lambda}_{l-1}\left[\phi(\hat{y}_{l-1}) - \hat{\ell}_k\right] + 2\hat{\Lambda}_k\hat{g}_k + \frac{\hat{\Lambda}_k\varepsilon}{2} + D^2
\end{aligned}$$

where the last inequality follows from the definition of D in (4) and the fact that $u_k^* \in \mathrm{cl}\,(\mathrm{dom}\, h)$. After simple algebraic manipulation, the above inequality implies that

$$\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l[\phi(\hat{y}_l) - \hat{\ell}_k] \leq \hat{\lambda}_{\lceil k/2\rceil-1}[\phi(\hat{y}_{\lceil k/2\rceil-1}) - \hat{\ell}_k] + 2\hat{\Lambda}_k\hat{g}_k + \frac{\hat{\Lambda}_k\varepsilon}{2} + D^2$$

$$\leq \hat{\lambda}_0[\phi(\hat{x}_0) - \hat{\ell}_0] + 2\hat{\Lambda}_k\hat{g}_k + \frac{\hat{\Lambda}_k\varepsilon}{2} + D^2 \tag{69}$$

where the last inequality uses the facts that the sequences $\{\hat{\lambda}_k\}$ and $\{\phi(\hat{y}_k)\}$ are non-increasing, and $\{\hat{\ell}_k\}$ is non-decreasing, in view of statements (b) and (c) of Lemma 4.3. Inequality (66) now follows by dividing the above inequality by $\hat{\Lambda}_k$ and using the definitions of $\hat{\Lambda}_k$ and $\hat{\Phi}_k$ in (56).

Now we prove (67). Using the definition of $\hat{g}_k$ in (56) and the fact that $\beta_l \leq \beta_{\lceil k/2\rceil}$ for every $l \geq \lceil k/2\rceil$ due to Lemma 4.3(c), we conclude that

$$\hat{g}_k \leq \frac{\beta_{\lceil k/2\rceil}}{\hat{\Lambda}_k} \sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l \left[\hat{\ell}_k - \hat{\ell}_{l-1}\right] \leq \beta_{\lceil \frac{k}{2}\rceil} \left[\phi_* - \hat{\ell}_0\right],$$

where the last inequality is due to (57) and statement (c) of Lemma 4.3, and hence that (67) holds. ∎

We now provide some remarks about Lemma 4.4.

First, we discuss the role of Step 3.e of Ad-GPB in light of this result. Relation (66) is analogous to (48), except that d_0 in (48) is upper bounded by the diameter D in (66), and most importantly, the additional term $\hat{g}_k$ is added to (66). Except this term, all other terms on the right-hand side of (66) can be bounded in the same way as in (48). Hence, the issue is how to nicely bound $\hat{g}_k$. To this end, we will use (67) to show that the strategy described in step 3.e of Ad-GPB, i.e., halving β_{k+1} whenever $\hat{g}_k$ is too large compared to the primal-dual gap $\hat{\Phi}_k - \hat{\ell}_k$, nicely bounds $\hat{g}_k$ in the long term, in a delayed manner, and hence the primal-dual gap too.

Finally, in contrast to Ad-GPB* which allows λ_{i_k} to be chosen in the interval $[\hat{\lambda}_{k-1}, 2\hat{\lambda}_{k-1}]$ (see (20)), Ad-GPB chooses $\lambda_{i_k} = \hat{\lambda}_{k-1}$, which implies that $\{\hat{\lambda}_k\}$ is non-increasing (see Lemma 4.3(c)). This property, which is used in the second inequality of (69), makes the above proof simpler. However, it is possible to develop a variant of Ad-GPB that chooses λ_{i_k} in the interval $[\hat{\lambda}_{k-1}, 2\hat{\lambda}_{k-1}]$ subject to the condition that, for some constant $C > 0$, the inequality $\hat{\lambda}_k[\phi(\hat{x}_k) - \hat{\ell}_k] \leq C \max\{\hat{\lambda}_0[\phi(\hat{x}_0) - \hat{\ell}_0], D^2\}$ holds for every $k \geq 1$. It can be shown that such a choice of λ_{i_k} allow us to extend Lemma 4.4 and subsequent results to obtain a complete analysis of the variant outlined above. However, for the sake of shortness and simplicity, we omit the details.

Now we state a result that provides key bounds on the primal-dual gap $\hat{\Phi}_k - \hat{\ell}_k$.

Lemma 4.5 *For every cycle index k of Ad-GPB, the following statements hold:*

a) if $\beta_{k+1} = \beta_k$, then we have

$$\hat{\Phi}_k - \hat{\ell}_k \leq \frac{4\hat{\lambda}_0[\phi(\hat{x}_0) - \hat{\ell}_0]}{3\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l} + \frac{4D^2}{3\sum_{l=\lceil k/2\rceil}^{k} \hat{\lambda}_l} + \frac{2\varepsilon}{3}; \tag{70}$$

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

$$\hat{\Phi}_k - \hat{\ell}_k \leq 8\hat{g}_k, \tag{71}$$

where $\hat{\Phi}_k$ is as in (56).

Proof: a) The update rule for β_{k+1} in step 3.e of Ad-GPB and the assumption that $\beta_{k+1} = \beta_k$ imply that $\hat{g}_k \leq (\hat{\Phi}_k - \hat{\ell}_k)/8$. This observation and inequality (66) then immediately imply (70).

b) Statement b) follows from the assumption that $\beta_{k+1} = \beta_k/2$ and the update rule for β_{k+1} in step 3.e of Ad-GPB. ∎

We now provide some remarks on Lemma 4.5. First, inequality (70) is analogous to (48). Second, if (70) holds at every iteration, then it can be shown, using similar arguments to those in §3.3.2, that the number of cycles generated by Ad-GPB can be nicely bounded. However, since (71) may also occur, the analysis in this section (especially, the proof of Theorem 4.1 below) is more involved than that for Ad-GPB* in §3.3.2.

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1 To simplify notation, let $\bar{K} = \bar{K}(\varepsilon)$. It is easy to see that

$$\frac{D^2 + \hat{\lambda}_0(\phi(\hat{x}_0) - \hat{\ell}_0)}{\bar{K}} \left(\frac{1}{\underline{\lambda}(\varepsilon)} + \frac{1}{\hat{\lambda}_0}\right) \leq \frac{\varepsilon}{8}, \qquad \frac{1}{2^{\bar{K}-3}} \left(\phi_* - \hat{\ell}_0\right) < \frac{\varepsilon}{\beta_1}. \tag{72}$$

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

$$\phi(\hat{y}_k) - \hat{\ell}_k > \varepsilon \quad \forall k \in \{1, \ldots, 4\bar{K}\}. \tag{73}$$

We then have that

$$\beta_{k+1} = \frac{\beta_k}{2} \quad \forall k \in \{\bar{K}, \ldots, 4\bar{K}\} \tag{74}$$

since otherwise we would have some $\beta_{k+1} = \beta_k$ for some $k \in \{\bar{K}, \ldots, 4\bar{K}\}$, and this together with (73) and (70), Lemma 4.2(b), and Lemma 4.3(b), would yield the contradiction that

$$\begin{aligned}
\varepsilon < \phi(\hat{y}_k) - \hat{\ell}_k &\overset{\mathbf{L}.4.3(b)}{\leq} \hat{\Phi}_k - \hat{\ell}_k \overset{(70)}{\leq} \frac{4\left(D^2 + \lambda_1(\phi(\hat{x}_0) - \hat{\ell}_0)\right)}{3\sum_{l=\lceil k/2 \rceil}^{k} \hat{\lambda}_l} + \frac{2\varepsilon}{3} \overset{\mathbf{L}.4.2(b)}{\leq} \frac{8\left(D^2 + \hat{\lambda}_0(\phi(\hat{x}_0) - \hat{\ell}_0)\right)}{3k \min\{\underline{\lambda}(\varepsilon), \hat{\lambda}_0\}} + \frac{2\varepsilon}{3} \\
&\leq \frac{8\left(D^2 + \hat{\lambda}_0(\phi(\hat{x}_0) - \hat{\ell}_0)\right)}{3\bar{K}} \left(\frac{1}{\underline{\lambda}(\varepsilon)} + \frac{1}{\hat{\lambda}_0}\right) + \frac{2\varepsilon}{3} \overset{(72)}{\leq} \frac{\varepsilon}{3} + \frac{2\varepsilon}{3} = \varepsilon,
\end{aligned}$$

where the last inequality is due to the first inequality in (72). Relations (67), (71), (73) and (74), and Lemma 4.3(b), all with $k = 4\bar{K}$, then yield

$$\varepsilon \overset{(73)}{<} \phi(\hat{y}_{4\bar{K}}) - \hat{\ell}_{4\bar{K}} \overset{\mathbf{L}.4.3(b)}{\leq} \hat{\phi}^a_{4\bar{K}} - \hat{\ell}_{4\bar{K}} \overset{(71)}{\leq} 8\hat{g}_{4\bar{K}} \overset{(67)}{\leq} 8\beta_{2\bar{K}} \left(\phi_* - \hat{\ell}_0\right) \overset{(74)}{\leq} \frac{1}{2^{\bar{K}-3}} \beta_{\bar{K}} \left(\phi_* - \hat{\ell}_0\right) \overset{(72)}{<} \frac{\beta_{\bar{K}}}{\beta_1} \varepsilon$$

where the last inequality is due to the second inequality in (72). Since $\beta_{\bar{K}} \leq \beta_1$, 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. Statements (a) and (c) of Lemma 4.2 then imply that the total number of iterations that Ad-GPB finds an iterate $\hat{y}_k$ satisfying $\phi(\hat{y}_k) - \hat{\ell}_k \leq \varepsilon$ is bounded by

$$\begin{aligned}
\sum_{k=1}^{\bar{k}} |\mathcal{C}_k| &\leq \sum_{k=1}^{\bar{k}} \left(\frac{2}{1-\tau} \ln^+ \left(\frac{\bar{t}}{\varepsilon}\right) + 2 + s_k\right) \\
&\leq \log_2^+ \frac{\hat{\lambda}_0}{\underline{\lambda}(\varepsilon)} + 4\bar{K} \left(\frac{2}{1-\tau} \ln^+ \left(\frac{\bar{t}}{\varepsilon}\right) + 2\right)
\end{aligned}$$

and hence by (59), due to the definitions of $\underline{\lambda}(\varepsilon)$ and $\bar{C} = \bar{C}(M, L, \hat{\lambda}_0, \tau, \varepsilon)$ in (35) and (23), respectively, and the fact that $1 \leq 1/(1-\tau)$. ∎

5 Computational experiments

This section reports the computational results of Ad-GPB* and Ad-GPB against other bundle methods. It contains three subsections. The first one presents computational results of these methods for solving

l_1 feasibility problems. The second one showcases their computational results for solving Lagrangian cut subproblems that are used to generate cuts in the context of integer programming. Finally, the third one presents their computational results for solving constrained l_1 feasibility problems.

We now describe all the methods that are used in our benchmark. We start with the ones that require the knowledge of the optimal value.

1. *Polyak subgradient (P-Sub*):* given x_k, this method computes

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

   where

$$\lambda_{\mathrm{pol}}(x) := \frac{\phi(x) - \phi_*}{\|g(x)\|^2} \tag{75}$$

   and $g(x) \in \partial f(x)$.

2. *GPB*:* this method is a special case of the one outlined in §2.2, and formally described in [22, 23], where the prox stepsize λ is set to $\lambda_{\mathrm{pol}}(x_0)$. This method requires a parameter τ.

3. *Ad-GPB*:* a cycle of Ad-GPB* (or Ad-GPB) is called good if the prox stepsize does not change within it. For any $k \geq 2$, Ad-GPB* either sets $\lambda_{i_k} = 2\hat{\lambda}_{k-1}$ if all the cycles before the k-th one are good, or else sets $\lambda_{i_k} = \hat{\lambda}_{k-1}$. This method requires parameters τ and $\hat{\lambda}_0$.

4. *P-Ad-GPB*:* this is a Polyak-type variant of Ad-GPB* where the initial prox stepsize for the kth-cycle is set to $\lambda_{i_k} = \alpha\lambda_{\mathrm{pol}}(\hat{x}_{k-1})$ for $k \geq 1$ and some $\alpha > 0$. This method requires a parameter τ.

5. *PB*:* this is a classical proximal bundle method with variable stepsize which is described in [13]. This method requires four parameters, namely, β, a, and λ_{min}, as in [13] and initial stepsize $\hat{\lambda}_0$.

6. *BG*:* This method is a variant of the classical proximal bundle method of [6] that sets the stepsize ρ_k for the k-th cycle as

$$\rho_k = \frac{f(\hat{x}_{k-1}) - f_*}{\|\hat{x}_{k-1} - x_*\|}. \tag{76}$$

   This method requires parameter β as in [6].

7. *APL*:* this is a variant of the Bundle Level (BL) method presented in [17] with α_k chosen as $\alpha_k = 2/(k+1)$. Moreover, it sets the lower bound to the known optimal value rather than updating it by solving a linear program at each iteration. This method requires two parameters, namely, β and θ as in [17].

We now describe the initialization and bundle update rules used by the methods. First, methods 3 and 5 choose the initial stepsize the same as in method 2. Second, methods 2, 3, 4, 5, and 6 update the bundle Γ_k using the two-cut scheme described in §3.2. The APL* method also employs a two-cut update, but in a slightly different manner from that described in §3.2. Third, methods 2, 3, and 4 set $\tau = 0.95$. Finally, all the aforementioned algorithms terminate when the primal gap becomes sufficiently small.

We now make several remarks on Algorithms 5–7. First, PB* relies on ϕ_* in two aspects: (i) its initial stepsize requires knowledge of ϕ_*, and (ii) its termination criterion depends on ϕ_*. Second, the stepsize choice (76) used for BG* requires knowledge of x_*. Even though it is completely impractical, we still use it in our benchmark since the optimal solutions for our randomly generated instances were all known. A more realistic approach would be to replace $\|\hat{x}_{k-1} - x_*\|$ by a uniform bound (e.g., a bound on the diameter of $\operatorname{dom} h$), but it was not clear how to obtain such a bound in the setting of the unconstrained problems used in our benchmark. Third, each iteration of APL in [17] generates a lower bound by solving a linear program, which is used to define a certain level set, and subsequently the next iterate. In contrast, APL* uses ϕ_* as the lower bound, which usually yields a different level set and iterate than those generated by APL.

We now describe the algorithms that do not require knowledge of the optimal value:

8. *Ad-GPB:* this method is described in §4. It chooses $\tau = 0.95$ and updates the bundle Γ_k according to the two-cut scheme described in §3.2. This method sets $\hat{\lambda}_0 = (\phi(x_0) - \hat{\ell}_0)/\|g(x_0)\|^2$ where $g(x_0) \in \partial f(x_0)$ and $\hat{\ell}_0 := \min\{\tilde{\ell}_f(x; x_0) : e^T x \leq D,\, x \geq 0\}$.

9. APL: this method is described in [17], which updates the bundle Γ_k using two cuts and chooses α_k as $\alpha_k = 2/(k+1)$. This method requires two parameters, namely, β and θ as in [17].

Both of algorithms 8 and 9 terminate when the primal-dual gap becomes small.

All experiments were written in MATLAB 2023a and were performed on PACE [1] with Dual Intel Xeon Gold 6226 CPUs @ 2.7 GHz (24 cores/node). The implementation of our algorithms is available at `https://github.com/Honghao-Zhang1/Parameter-free-proximal-bundle-methods-with-adaptive-stepsizes-.git`.

We now discuss how the required parameters (if any) for the above algorithms were tuned in our benchmark. First, the tuning of the parameter α in P-Ad-GPB* is performed only on small instances of Problem 1. The resulting choice is then fixed and used uniformly across all other problem instances without further adjustment. Specifically, α is selected from the set $\{1, 20, 40, 60\}$. Among these candidates, $\alpha = 40$ consistently provides the best performance and is therefore used in all reported experiments.

Second, for the parameter tuning of APL and APL*, pairs (β, θ) such that $\beta \in \{0.3, 0.5, 0.7, 0.9\}$ and $\theta \in \{0.2, 0.35, 0.5, 0.65\}$ were used. For PB*, we used triples $(\beta, a, \lambda_{min})$ such that $\beta \in \{0.1, 0.5, 0.7\}$, $\lambda_{\min} \in \{10^{-6}, 10^{-5}, 10^{-3}\}$, and $a \in \{2, 4, 5\}$. For PB*, we used scalars $\beta \in \{0.1, 0.5, 0.7\}$. For all these methods, the best combination of parameters was selected using small representative instances from each problem class, and then used to solve all instances of the same problem class in our benchmark.

5.1 l_1 feasibility problem

The main goal of this subsection is to benchmark the algorithms that require knowledge of ϕ_* on the l_1 feasibility problem.

We are now ready to describe the l_1 feasibility problem. The problem can be formulated as:

$$f_* := \min_{x \geq 0} f(x) := \|Ax - b\|_1 \tag{77}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^n$ are known. We now describe the way of generating the data. 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 $\mathcal{U}[0, 1000]$. Vector b is then 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 (77), whose optimal value f_* equals zero.

To check how sensitive GPB* and Ad-GPB* are relative to the initial choice of prox stepsize $\hat{\lambda}_0$, we test them for $\hat{\lambda}_0 = \alpha\lambda_{\text{pol}}(x_0)$ where $\alpha \in \{0.01, 1, 100\}$. The results are presented in Table 1. 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. For each instance, the row labeled "CPU/it" reports the total CPU time and the average time per iteration, where the latter is scaled by 10^{-3}. The row labeled "O/AI" reports the numbers of outer and average inner iterations, respectively. An entry marked $*$ indicates that the method fails to achieve the desired relative accuracy within the time limit; in this case, the corresponding "O/AI" entry reports the achieved relative accuracy as defined in (78) below. Bold numbers highlight the best-performing method for each instance.

All the methods tested in this subsection are terminated based on the following criterion:

$$f(x_k) - f_* \leq \bar{\varepsilon}[f(x_0) - f_* + 1] \tag{78}$$

where $\bar{\varepsilon} = 10^{-4}$.

[1] https://pace.gatech.edu/

ALG.		GPB*			Ad-GPB*		
$(\theta_m, \theta_n, \theta_s)\backslash\alpha$		10^{-2}	1	10^2	10^{-2}	1	10^2
$(1, 20, 10^{-2})$	CPU/it	194/2.84	464/3.02	$*$	38/2.07	44/2.08	**17**/2.06
	O/AI	9924/6.88	102/1507.4	$.13e{-}03$	9152/2.01	10607/2.01	4002/2.02
$(3, 30, 10^{-2})$	CPU/it	809/5.53	691/5.72	$*$	239/4.45	**157**/4.44	221/4.42
	O/AI	12242/11.95	123/982.65	$.13e{-}03$	26802/2.00	17647/2.00	24975/2.00
$(5, 50, 10^{-2})$	CPU/it	1561/12.65	1273/12.80	$*$	631/11.42	**420**/11.56	459/10.77
	O/AI	10171/12.13	106/937.76	$.13e{-}03$	27607/2.00	18140/2.00	21251/2.00
$(10, 100, 10^{-3})$	CPU/it	1558/10.24	1373/10.39	$*$	489/8.21	**367**/8.09	474/8.19
	O/AI	13260/11.48	134/985.78	$.13e{-}03$	29776/2.00	22642/2.00	28888/2.00
$(20, 200, 10^{-3})$	CPU/it	3472/25.49	2902/26.07	$*$	1573/22.28	**917**/22.32	1308/22.30
	O/AI	11733/11.61	120/927.34	$.13e{-}03$	35266/2.00	20511/2.00	29282/2.00
$(30, 300, 10^{-3})$	CPU/it	$*$	$*$	$*$	3051/47.56	**1850**/48.28	2590/47.86
	O/AI	$.28e{-}03$	$.16e{-}03$	$.13e{-}03$	32046/2.00	19131/2.00	27009/2.00

Table 1: Numerical results for sparse instances. The stopping criterion (78) with $\bar{\varepsilon} = 10^{-4}$ is used, and a time limit of 3600 seconds (1 hour) is imposed.

Table 1 demonstrates that Ad-GPB* outperforms GPB* and exhibits greater robustness with respect to the initial stepsize. Moreover, Ad-GPB* require substantially fewer inner iterations than GPB, confirming the theoretical insight that these variants adaptively adjust stepsizes within each cycle in an efficient manner.

Next, we compare methods that require the optimal value on a larger set of instances. The computational results for ten sparse instances are reported in Table 2, where the target tolerance $\bar{\varepsilon}$ in (78) is fixed at 10^{-4}. The quantities θ_m, θ_n, and θ_s are defined as in Table 1. Each table entry corresponds to the CPU running time in seconds. An entry marked $*/N$ indicates that the computed solution failed to achieve the desired relative accuracy within the time limit, in which case N denotes the relative accuracy of the computed solution as defined in (78). Bold numbers highlight the method achieving the best performance for each instance. The time limit for all runs reported in Table 2 is one hour.

$(\theta_m, \theta_n, \theta_s)$	Ad-GPB*	P-Ad-GPB*	P-Sub*	BG*	PB*	GPB*	APL*
$(1,20,10^{-2})$	44	21	482	*/.45e-02	**3**	464	*/.30e-02
$(2,20,10^{-2})$	73	**56**	444	*/.30e-02	835	357	*/.23e-02
$(3,30,10^{-2})$	156	**85**	1109	*/.31e-02	*/.14e-03	690	*/.19e-02
$(4,40,10^{-2})$	385	**147**	274 4	*/.33e-02	*/.14e-03	1213	*/.17e-02
$(5,50,10^{-2})$	403	**163**	3276	*/.34e-02	*/.10e-03	1271	*/.15e-02
$(10,100,10^{-3})$	367	**143**	2521	*/.35e-02	*/.16e-03	1337	*/.11e-02
$(15,150,10^{-3})$	703	**290**	*/.14e-03	*/.38e-02	*/.15e-03	2206	*/.87e-03
$(20,200,10^{-3})$	886	**449**	*/.16e-03	*/.39e-02	*/.13e-03	2885	*/.78e-03
$(25,250,10^{-3})$	1442	**560**	*/.23e-03	*/.43e-02	*/.14e-03	*/.12e-03	*/.70e-03
$(30,300,10^{-3})$	1810	**714**	*/.27e-03	*/.38e-02	*/.13e-03	*/.15e-03	*/.64e-03

Table 2: Numerical results for sparse instances. Stopping criterion (78) with $\bar{\varepsilon} = 10^{-4}$ is used and a time limit of 3600 seconds (1 hour) is given. Each entry reports CPU running time in seconds. An entry marked $*/N$ indicates that the computed solution failed to achieve the desired relative accuracy within the time limit, where N denotes the attained relative accuracy.

Tables 2 demonstrate that Ad-GPB* and P-Ad-GPB* generally outperform other methods in terms of CPU running time. Additionally, P-Ad-GPB* stands out as a particularly effective variant, outperforming other methods in nine out of ten instances.

5.2 Lagrangian cut problem

This subsection presents the numerical results compare Ad-GPB* and P-Ad-GPB* against P-Sub* and APL* on a convex nonsmooth optimization problem that has broad applications in the field of integer programming (see e.g. [32]). For completeness, the results of PB are also included.

The problem considered in this subsection arises in the context of solving the the stochastic binary multi-knapsack problem described next. Let Ξ be a set with a finite number of scenarios, and let $q : \Xi \to \mathbb{R}^n$, $(d, T, W) \in \mathbb{R}^m \times \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n}$ and $(b, A) \in \mathbb{R}^l \times \mathbb{R}^{l\times n}$ be given. The problem is

$$\begin{aligned} \min_x \ & c^T x + P(x) \\ \text{s.t. } & Ax \geq b \\ & x \in \{0,1\}^n \end{aligned} \tag{79}$$

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

$$\begin{aligned} P_\xi(x) := \min_y \ & q(\xi)^T y \\ \text{s.t. } & Wy \geq d - Tx \\ & y \in \{0,1\}^n \end{aligned} \tag{80}$$

for every $x \in \{0,1\}^n$. In the second-stage problem (80), only the objective vector $q(\xi)$ is stochastic.

Benders decomposition (see e.g. [8, 30]) is an efficient cutting-plane approach for solving (79) 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

$$\begin{aligned} x_k = \operatorname{argmin}_x \ & c^T x + P_k(x) \\ \text{s.t. } & Ax \geq b \\ & x \in \{0,1\}^n \end{aligned} \tag{81}$$

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 (81) 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 $\xi \in \Xi$, (80) is equivalent to

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

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

$$D_\xi(x) := \max_\pi L_\xi(x;\pi) \tag{82}$$

where

$$\begin{aligned} L_\xi(x;\pi) := \min_{y,u} \ & q(\xi)^T y - \pi^T(u - x) \\ \text{s.t. } & Wy + Tu \geq d \\ & y \in \{0,1\}^n,\ u \in [0,1]^n. \end{aligned} \tag{83}$$

Let $\pi_\xi(x)$ denote an optimal solution of (82). The optimal values $P_\xi(\cdot)$ and $D_\xi(\cdot)$ of (80) and (82), respectively, are known to satisfy the following two properties for every $\xi \in \Xi$ and $x \in \{0,1\}^n$:

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

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

Property (i) and the inequality $P_\xi(x) \geq D_\xi(x)$ for every $x \in \{0,1\}^n$ can be found in many textbooks dealing with Lagrangian duality theory. On the other hand, property (ii) has been established in [32] by exploiting the special binary structure of (80). 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 (80) can be efficiently computed for every $\xi \in \Xi$. Computation of $\pi(x)$ assumes that an optimal solution of (82) can be computed for every $\xi \in \Xi$. Noting that (82) 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 (80), we use the Ad-GPB* variant of Ad-GPB to obtain a near optimal solution $\approx \pi_\xi(x)$ of (82). For the purpose of this subsection, we use several instances of (82) 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 (82) 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 (83). Moreover, if $(u_\xi(x;\pi), y_\xi(x;\pi)))$ denotes an optimal solution of (83), then $u_\xi(x;\pi)$ yields a subgradient of $-L_\xi(x, \cdot)$ at π. It is worth noting (82) is a non-smooth convex problem that does not seem to be tractable by the methods discussed in the papers (see e.g. [4, 10, 11, 16, 26, 27, 28, 29]) for solving min-max smooth convex-concave saddle-point problems, mainly due to the integrality condition imposed on the decision variable y in (83).

Next we describe how the data of (79) and (80) is generated. We generate three random instances of (79), 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 $d = 3(W\mathbf{1} + T_1\mathbf{1})/4$ where $\mathbf{1}$ denotes the vector of ones.

For each randomly generated instance of (79), 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 (82). Hence, each randomly generated instance of (79) yields a total of sixty LD instances as in (82). The total time to solve these sixty LD instances are given in Table 3 for the three instances of (79) (named $I1$, $I2$ and $I3$ in the table) and all the benchmarked methods considered in this section. In this comparison, Ad-GPB*, P-Ad-GPB* and PB set the initial stepsize $\hat{\lambda}_0$ to $\lambda_{\text{pol}}(\pi_0)$ where the entries of π_0 are i.i.d. generated from the uniform distribution in $(0, 1)$.

All the methods tested are terminated based on the following criterion:

$$L_\xi(x;\pi_k) - L^*_\xi \leq \bar{\varepsilon} \tag{84}$$

where $\bar{\varepsilon} = 10^{-4}$.

	Ad-GPB*	P-Ad-GPB*	P-Sub*	PB*	APL*
$I1$	144s	**16s**	431s	75s	7450s
$I2$	214s	**16s**	1175s	82s	20731s
$I3$	213s	**16s**	1167s	82s	20650s

Table 3: Numerical results for solving LD subproblems. Stopping criterion (84) with $\bar{\varepsilon} = 10^{-4}$ is used.

Table 3 shows that, similar to Table 2, P-Ad-GPB* outperforms Ad-GPB*, which in turn outperforms APL*.

5.3 Constrained l_1 problem

This subsection reports the computational results of Ad-GPB against APL on a constrained l_1 feasibility problem.

The constrained l_1 feasibility problem consists of

$$\begin{aligned} \min f(x) &:= \|Ax - b\|_1 \\ \textbf{s.t. } & e^T x \leq D \\ & x \geq 0 \end{aligned} \tag{85}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^n$, and $D \in \mathbb{R}^{++}$ are known. 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 $\mathcal{U}[0,1000]$. Vector b is radnomly generated in the form $b = (b_0)^2$ where the entries of b_0 are i.i.d. sampled from the standard Normal distribution $\mathcal{N}(0,1)$. Finally, we generated $x_0 = \mathbf{1}/D$. The two benchmark methods are tested on eighteen randomly generated sparse constrained l_1 feasibility problems and are terminated when the primal-dual criterion

$$f(x_k) - \hat{\ell}_k \leq \bar{\varepsilon}(f(x_0) - \hat{\ell}_0) \tag{86}$$

is satisfied, where $\hat{\ell}_k$ is a lower bound on ϕ_* and $\hat{\ell}_0 := \min\{\tilde{\ell}_f(x; x_0) : e^T x \leq D,\, x \geq 0\}$.

We now describe some details about all the tables that appear in this subsection. We set the target $\bar{\varepsilon}$ in (86) as 10^{-3} . The quantities θ_m, θ_n and θ_s are defined as in Table 1. Each table entry corresponds to the CPU running time in seconds. An entry marked $*/N$ indicates that the computed solution failed to achieve the desired relative accuracy within the time limit, in which case N denotes the relative accuracy of the computed solution as defined in (78). Bold numbers highlight the method achieving the best performance for each instance.

$(\theta_m, \theta_n, \theta_s, D)$	Ad-GPB	APL	$(\theta_m, \theta_n, \theta_s, D)$	Ad-GPB	APL
(1,20,10^{-2},100)	**23**	*/.97e+00	(1,20,10^{-2},50)	**25**	*/.80e+00
(1.5,15,10^{-2},100)	**253**	*/.97e+00	(1.5,15,10^{-2},50)	**421**	*/.97e+00
(2,20,10^{-2}, 100)	**71**	*/.98e+00	(2,20,10^{-2}, 50)	**46**	*/.85e+00
(2.5,25,10^{-2}, 100)	**460**	*/.98e+00	(2.5,25,10^{-2}, 50)	**466**	*/.97e+00
(3,30,10^{-2},100)	**304**	*/.96e+00	(3,30,10^{-2},50)	**218**	*/.65e+00
(3.5,35,10^{-2},100)	**60**	*/.96e+00	(3.5,35,10^{-2},50)	**146**	*/.96e+00
(4,40,10^{-2},100)	**962**	*/.80e+00	(4,40,10^{-2},50)	**319**	*/.54e+00
(4.5,45,10^{-2},100)	**341**	*/.92e+00	(4.5,45,10^{-2},50)	**375**	*/.80e+00
(5,50,10^{-2},100)	**680**	*/.76e+00	(5,50,10^{-2},50)	**996**	*/.51e+00

Table 4: Numerical results for sparse instances. Stopping criterion (86) with $\bar{\varepsilon} = 10^{-3}$ is used and a time limit of 1800 seconds (half an hour) is given. Each table entry corresponds to the CPU running time in seconds. An entry marked $*/N$ indicates that the computed solution failed to achieve the desired relative accuracy within the time limit, in which case N denotes the relative accuracy of the computed solution as defined in (86).

Table 4 shows that Ad-GPB significantly outperforms APL.

6 Concluding remarks

This paper presents a parameter and line-search 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 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 complexity of P-Ad-GPB* described in §5 as it 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 [23]) since it is not a special case of (18) as already observed in the remark at the end of §3.2. It would be interesting to extend the analysis of this paper to establish the complexity of Ad-GPB* and 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] A. Ben-Tal and A. Nemirovski. Non-euclidean restricted memory level method for large-scale convex optimization. *Mathematical Programming*, 102(3):407–456, 2005.

[3] 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.

[4] 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.

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

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

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

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

[9] 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.

[10] 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.

[11] 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.

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

[13] K. C. Kiwiel. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Mathematical Programming*, 69(1-3):89–109, 1995.

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

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

[16] 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.

[17] G. Lan. Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization. *Mathematical Programming*, 149(1-2):1–45, 2015.

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

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

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

[21] Claude Lemaréchal, Arkadii Nemirovskii, and Yurii Nesterov. New variants of bundle methods. *Mathematical programming*, 69(1):111–147, 1995.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

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

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