Math. Program., Ser. B (2014) 148:241–277
DOI 10.1007/s10107-014-0809-6

FULL LENGTH PAPER

Convex proximal bundle methods in depth: a unified analysis for inexact oracles

W. de Oliveira · C. Sagastizábal · C. Lemaréchal

Received: 5 February 2013 / Accepted: 16 August 2014 / Published online: 7 September 2014

Abstract The last few years have seen the advent of a new generation of bundle methods, capable to handle inexact oracles, polluted by "noise". Proving convergence of a bundle method is never simple and coping with inexact oracles substantially increases the technicalities. Besides, several variants exist to deal with noise, each one needing an ad hoc proof to show convergence. We state a synthetic convergence theory, in which we highlight the main arguments and specify which assumption is used to establish each intermediate result. The framework is comprehensive and generalizes in various ways a number of algorithms proposed in the literature. Based on the ingredients of our synthetic theory, we consider various bundle methods adapted to oracles for which high accuracy is possible, yet it is preferable not to make exact calculations often, because they are too time consuming.

Mathematics Subject Classification 90C · 49M · 65K

1 Introduction and general aim

We are interested in the problem

$$\min f(u), \quad u \in \mathbb{R}^n, \tag{1.1}$$

W. de Oliveira
Instituto Nacional de Matemática Pura e Aplicada—IMPA, Rio de Janeiro, Brazil
e-mail: wlo@impa.br

C. Sagastizábal (✉)
IMPA, Rio de Janeiro, Brazil
e-mail: sagastiz@impa.br

C. Lemaréchal
Inria, 655 avenue de l'Europe, Montbonnot, 38334 Saint Ismier, France
e-mail: claude.lemarechal@inria.fr

where $f(\cdot): \mathbb{R}^n \to \mathbb{R}$ is a (finite-valued) convex function. For each given $u \in \mathbb{R}^n$, an oracle—a noisy one-delivers inexact information, namely

$$\begin{cases} f_u = f(u) - \eta_u \quad \text{and} \\ g_u \in \mathbb{R}^n \text{ such that } f(\cdot) \geqslant f_u + \langle g_u, \cdot - u\rangle - \eta_u^g \\ \text{with } \eta_u \leqslant \eta \text{ and } \eta_u^g \leqslant \eta \quad \text{for all } u \in \mathbb{R}^n \end{cases} \tag{1.2}$$

where the error bound η is possibly unknown. For a given u, function values will be denoted by a letter such as f_u; reserving the notation $f(\cdot)$ for the function itself.

To solve (1.1) we put in place a framework for bundle methods dealing with oracles of the type (1.2). The considered setting is versatile and general, in the sense that it covers and extends previous literature, such as the inexact bundle methods [12,15, 20], the incremental bundle method [6], and the partly inexact method [16]. We also consider new methods, the Controllable Bundle Algorithm 5.4 and the Asymptotically Exact Algorithm in Sect. 7.1.4. The latter is a proximal variant of the level bundle method for oracles with on-demand accuracy considered in [4].

Similarly to the spirit of [3], our development highlights the main arguments and assumptions used to establish each intermediate result. The analysis is presented in a way that reveals how different procedures controlling oracle noise result in algorithms solving (1.1) with different degrees of accuracy.

To give a flavor of different situations fitting (1.2), Sect. 2 starts with a broad set of examples. Section 3 organizes the essential features of bundle methods in two sets of *parameters* whose particularization gives rise to specific algorithms. The parametric setting is useful to state a general algorithmic pattern in Sect. 4. The important mechanism of *noise attenuation*, to be put in place when the oracle error cannot be controlled, is addressed in Sect. 5. This section also illustrates a parameter specification for the algorithmic pattern, the Controllable Bundle Algorithm 5.4, used along Sect. 6 to guide the reader through the various convergence results therein. The final Sect. 7 considers many algorithms covered by our synthetic theory, including constrained bundle methods.

2 Oracle examples

By the second line in (1.2),

$$f(\cdot) \geqslant f(u) + \langle g_u, \cdot - u\rangle - (\eta_u + \eta_u^g) \tag{2.1}$$

from which, evaluating at u we deduce that, independently of the errors sign, $\eta_u + \eta_u^g \geqslant 0$. As a result, g_u is a Convex Analysis ε-subgradient:

$$g_u \in \partial_{\eta_u + \eta_u^g} f(u) \quad \text{with } \eta_u + \eta_u^g \geqslant 0 \quad \text{for all } u \in \mathbb{R}^n. \tag{2.2}$$

Even if in (1.2) the value of the upper error bound η is unknown, the inequality above implies that $\eta \geqslant \eta_u \geqslant -\eta_u^g \geqslant -\eta$: both oracle errors are bounded from below by $-\eta$.

An *exact oracle* has $\eta_u \equiv \eta_u^g \equiv 0$, the output is $f_u = f(u)$ and a true subgradient. In an important subclass of inexact oracles illustrated by Examples 2.1 and 2.2, $\eta_u^g \equiv 0$ and $\eta_u \geqslant 0$, by (2.2). Following [21], we shall call this subclass of *lower oracles*, because a lower linearization of $f(\cdot)$ is available:

$$f(u) - \eta_u = f_u \leqslant f(u) \quad \text{and} \quad f(\cdot) \geqslant f_u + \langle g_u, \cdot - u \rangle \, . \tag{2.3}$$

Upper oracles, by contrast, can over-estimate function values: in (2.1) the error η^g is positive.

Example 2.1 (Minimax: Lagrangian) For given functions $h(\cdot)$ and $c(\cdot)$ and X a non-empty compact set, (1.1) is dual to the primal problem

$$\max_{x \in X} h(x), \quad c(x) = 0 \in \mathbb{R}^n. \tag{2.4}$$

Specifically, for a multiplier $u \in \mathbb{R}^n$ the dual function is given by

$$f(u) := \max_{x \in X} L(x, u), \quad \text{with } L(x, u) := h(x) + \langle u, c(x) \rangle \, .$$

In a more general setting $L(x, \cdot)$ is convex. Suppose that given u, the oracle outputs $f_u := L(x, u)$ for some $x \in X$, together with some $g_u \in \partial_u L(x, u)$. By convexity of $L(x, \cdot)$ and definition of $f(\cdot)$,

$$f_u + \langle g, \cdot - u_0 \rangle = L(x, u) + \langle g, \cdot - u_0 \rangle \leqslant L(x, \cdot) \leqslant f(\cdot).$$

For this lower oracle (2.3) holds. Such is the case in Lagrangian relaxation or column generation when the operation $\max_{x \in X} L(x, u)$ is not performed exactly. □

Example 2.2 (*Minimax: Two-Stage Stochastic Linear Programs*) Consider a stochastic linear program with decision variables organized in two levels, denoted by u and y for the first and second stage, respectively. If $\xi \in \Xi$ represents uncertainty, for vectors e and $q(\xi)$ and matrices $T(\xi)$ and W, the corresponding two-stage linear program with fixed recourse is

$$\begin{cases} \min_{u,y} & \langle e, u \rangle + \mathbb{E}[\langle q(\xi), y \rangle] \\ \text{s.t.} & T(\xi)u + Wy = d(\xi) \quad \text{for almost every } \xi \in \Xi, \\ & y \geqslant 0, \end{cases}$$

where we use the symbol $\mathbb{E}(\cdot)$ for the expected value. For fixed u and ξ the *recourse* function

$$Q(u; \xi) := \inf_{y \geqslant 0} \left\{ \langle q(\xi), y \rangle \text{ s.t. } Wy = d(\xi) - T(\xi)u \right\}$$

gives in (1.1) an objective of the form $f(u) := \langle e, u \rangle + \mathbb{E}[Q(u; \xi)]$, which is finite-valued when recourse is relatively complete. We now explain how to build different oracles for this type of problems.

A dumb lower oracle For each fixed u and a given realization ξ, the evaluation of the recourse function can be done by solving the dual linear program

$$Q(u;\xi) = \sup_x \Big\{ \langle d(\xi) - T(\xi)u, x\rangle \quad \text{s.t.} \quad W^\top x \leqslant q(\xi) \Big\}.$$

If, to speed up calculations, instead of performing the max-operation for the considered ξ we just take a feasible point $x_{u,\xi}$ (satisfying $W^\top x_{u,\xi} \leqslant q(\xi)$), then an oracle taking $f_u := \langle e, u\rangle + \mathbb{E}[\langle d(\xi) \ -T(\xi)u, x_{u,\xi}\rangle]$, and $g_u := e - \mathbb{E}[T(\xi)^\top x_{u,\xi}]$ is of lower type and fits (2.3).
A controllable lower oracle A better estimate can be computed by making some iterations of a primal-dual linear programming solver. The oracle receives as *additional input an error bound* $\bar\eta_u \geqslant 0$ and stops the primal-dual solver as soon as it finds a feasible point $x_{u,\xi}$ for which $\big\langle d(\xi) - T(\xi)u, x_{u,\xi}\big\rangle - Q(u;\xi) \leqslant \bar\eta_u$. For this oracle the subgradient error η_u^g is null and $f_u \in [f(u) - \bar\eta_u, f(u)]$ for any error bound $\bar\eta_u$ chosen by the user [4].
Asymptotically exact oracles To build an oracle that is eventually exact everywhere from the controllable oracle, just take $\bar\eta_u \to 0$, to force the error bound to vanish along iterations.

A smarter oracle, called *partly asymptotically exact*, requires eventual exactness only for some input points u_k. This is done by combining the three preceding lower oracles, as follows. Together with $(u, \bar\eta_u)$, the oracle receives *as additional input a target* γ_u and must compute f_u within the "on-demand" accuracy $\bar\eta_u$ only when the target is reached:

$$f_u \text{ is computed}$$
$$\begin{cases} \text{as in the dumb lower oracle } (f_u \in [f(u) - \eta, f(u)]) \text{ if } f_u > \gamma_u, \\ \text{as in the controllable lower oracle } (f_u \in [f(u) - \bar\eta_u, f(u)]) \text{ if } f_u \leqslant \gamma_u. \end{cases}$$

The bound $\eta \geqslant 0$ may be unknown while $\bar\eta_u \geqslant 0$ is known and controllable. In addition to the dumb and controllable oracles, a third one, asymptotically exact, comes into play when the user sets the target as a goal of decrease for f at u_k and drives $\bar\eta_{u_k}$ to zero; see [4].
An upper oracle Instead of considering all the random events (Ξ may be infinite), a small finite subset can be drawn from the sampling space for each given u. The recourse function is computed exactly only for ξ in that subset; the corresponding minimizers $x_{u,\xi}$ give approximate functional and gradient values in (2.3) whose errors have unknown sign. The bound η may be unknown but exists and depends on the probability distribution. □

Example 2.3 (*Chance-Constrained Programs*) For a probability level $p \in (0, 1]$, a simple convex and compact polyhedron U, and a log-concave probability distribution for ξ, in

$$\begin{cases} \min_{u\in U} & \langle h, u\rangle \\ \text{s.t.} & \mathbb{P}(Tu \leqslant \xi) \geqslant p \end{cases}$$

the constraint is convex and can be smooth, but its oracle requires computing a costly gradient. To allow for approximate calculations, [21] minimizes over U the *improvement function*

$$f(u) = \max\{\langle h, u\rangle - \tau_1, \, -\ln(p) - \ln[\mathbb{P}(Tu \leqslant \xi)] - \tau_2\} \text{ for a parameter } \tau \in \mathbb{R}^2;$$

see Sect. 7.4. The upper oracle in [21] delivers an unknown error that is bounded if so is U. The error can be driven to zero at the expense of heavy computations. □

Example 2.4 (Convex Composite Functions) All the functions above involve some maximization operation. In a more general setting, including eigenvalue optimization [11], given a convex function $h(\cdot)$ that is positively homogeneous (like the max-function) and a convex smooth operator $c(\cdot)$, the objective in (1.1) can have the form $f(\cdot) = (h \circ c)(\cdot)$. Suppose $c : \mathbb{R}^n \to \mathbb{R}^m$ and let $Dc(\cdot)$ denote its Jacobian. Given $\hat{u} \in \mathbb{R}^n$, the function $F(\cdot; \hat{u}) : \mathbb{R}^m \to \mathbb{R}$ defined by $F(\cdot; \hat{u}) := h(c(\hat{u}) + Dc(\hat{u})(\cdot - \hat{u}))$ is used in the *composite bundle method* [19] to solve (1.1). Since computing the Jacobian matrix is expensive, using oracle information for the convex function $F(\cdot; \hat{u})$ eases calculations. With respect to the true f-information, nothing can be said about the error sign: the oracle can be of upper type; see Sect. 7.5. □

We will retain from these oracles that several situations can occur:

- *Dumb oracle* not much is known about output f_u and g_u. Such is the case in Example 2.2. of the dumb lower and the (totally dumb) upper oracles.
- *Informative oracle* in addition to f_u and g_u, the oracle returns some reliable upper bound $\bar{\eta}_u$ for η_u. For Example 2.1, when the $L(\cdot, u)$-maximization is a constrained problem solved by a primal-dual method, the oracle outputs some feasible x_u, giving $f_u = L(x_u, u)$ and computes with dual arguments an upper bound M_u (used to define $\bar{\eta}_u := M_u - f_u \geqslant 0$).
- *Controllable oracle* in addition to u, the oracle receives $\bar{\eta}_u$ and must compute f_u within $\bar{\eta}_u$-accuracy. This situation has itself several sub-cases:
  - *Achievable* high accuracy. A smaller $\bar{\eta}_u$ implies an acceptable increase in computing times, as with the controllable lower oracle in Example 2.2.
  - *Partly achievable* high accuracy. It is acceptable to make exact calculations at "good" points, as in the partly asymptotically exact oracle in Example 2.2. By this token, $\bar{\eta}_u$ can be managed more aggressively. We shall see in Sect. 7.5 that the convex composite oracle in Example 2.4 fits this category.
  - *Exorbitantly* high accuracy. For instance if in Example 2.1 the $L(\cdot, u)$-maximization is a difficult, combinatorial and/or large scale problem, or when in Example 2.2 the sampling space is too large and/or the probability distribution too involved.

3 Parametric characterization of bundle methods

An algorithm to solve (1.1) should construct a sequence $\{(\hat{u}_k, \hat{g}_k)\}$ whose respective primal and dual terms aim at minimizing $f(\cdot)$ and approaching $0 \in \mathbb{R}^n$ (thus providing a certificate of optimality). The *centers* $\hat{u}_k$ defined below are extracted from the algorithm iterate sequence $\{u_k\}$ by collecting points that provide sufficient progress towards the goal of solving (1.1); for instance, by reducing the functional value. We describe this construction for proximal bundle methods with the least possible references to the oracle errors. Noise comes into play only *a posteriori*, to determine to which extent the algorithm really solves (1.1).

3.1 Defining the set **P**

To make the parametric notation clear, we start with the pure cutting-plane methods [2,14]. Having called the oracle at a number of points u_j, these algorithms accumulate linearizations

$$f_j^L(u) := f_{u_j} + \langle g_{u_j}, u - u_j \rangle. \tag{3.1}$$

To compute the next iterate, u_{k+1}, the *master-program* minimizes the cutting-plane model

$$\check{f}_k(\cdot) := \max\left\{ f_j^L(\cdot) \;:\; j \in J_k := \{1, \dots, k\}\right\}.$$

As a result, the following set of *parameters* fully characterizes a cutting-plane method:

$$\mathbf{P} = \left\{ \begin{array}{l} \text{the convex } \mathit{model}\ \check{f}_k(\cdot) \text{ and} \\ \text{a measure of progress, such as } f_{u_{k+1}} - \check{f}_k(u_{k+1}) \end{array} \right\},$$

that is, the optimality certificate used to stop the algorithm. Taking the maximum over J_k in the inequalities $f_j^L(\cdot) \leqslant f(\cdot) + \eta_{u_j}^g$, obtained from (1.2), gives:

$$\check{f}_k(\cdot) \leqslant f(\cdot) + \max_{j \in J_k} \eta_{u_j}^g \quad \text{for all } k. \tag{3.2}$$

For *stabilized* cutting-plane variants, such as [2,4,8,11], the set **P** includes some devices guaranteeing *descent* for special iterates, called *centers*:

$$\mathbf{P} = \left\{ \begin{array}{l} \text{a convex model } f_k^M, \text{ possibly different from the cutting-plane one,} \\ \text{a measure of progress to stop the algorithm,} \\ \text{a } \mathit{stability center}\ \hat{u}_k, \text{ a past iterate deemed ``good enough'',} \\ \text{a proximal stabilization } \frac{1}{2t_k}|\cdot - \hat{u}_k|^2, \\ \text{other parameters and updating rules, including } t_{k+1},\ f_{k+1}^M(\cdot), \text{ etc.} \end{array} \right\}.$$

The quadratic norm, or proximal stabilization, is replaced by more general terms in [7]. In all these methods, u_{k+1} is the unique minimum of a stabilized model function:

$$\min_{u \in \mathbb{R}^n} f_k^S(u), \quad \text{for } f_k^S(u) := f_k^M(u) + \frac{1}{2t_k}|u - \hat{u}_k|^2. \tag{3.3}$$

When the problem (1.1) is constrained by a simple polyhedron as in Example 2.3, the minimization in (3.3) incorporates this feasible set; see Sect. 7.3.

The model in **P** *does not need* to be one based on cutting planes, although the relation

$$f_k^M(\cdot) \leqslant \check{f}_k(\cdot) \quad \text{for all } k, \tag{3.4}$$

generally holds, for example when compressing the bundle J_k; see Sects. 5.2 and 7.1. Traditional bundle methods work with polyhedral approximations; in eigenvalue optimization, the spectral bundle methods [11,18] use non-polyhedral models. The main requirement on the choice of $f_k^M(\cdot)$ is pragmatic: problem (3.3) should be easily solvable. Relevant assumptions will be given later, as the need arises; for now we just mention that for lower oracles like in (2.3), taking $f_k^M(\cdot) := \check{f}_k(\cdot)$ gives

$$f_k^M(\cdot) \leqslant f(\cdot) \quad \text{for all } k. \tag{3.5}$$

When this inequality holds we shall say that a *lower model* is available.

3.2 Ensuring descent for the center subsequence: the set D

We single out from the set **P** the criterion to decide when an iterate becomes the next center, and write it as a rule depending on objects specified by a second set, checking if the iterate provides sufficient descent. The resulting set **D** is defined first for exact oracles.

3.2.1 Exact oracles

When the oracle calculations are exact, descent is determined by observing progress towards the goal of minimizing the objective function. Progress can be measured relative to either the model, or some nominal reference value, or the objective function itself. The three corresponding measures are respectively called and denoted by

$$\textit{model decrease } \delta_k^M, \textit{nominal decrease } \delta_k^N, \text{ and } \textit{effective decrease } \delta_k^E.$$

With exact oracles, the model decrease is $\delta_k^M = f(\hat{u}_k) - f_k^M(u_{k+1})$. The nominal decrease, a non-negative measure because (3.3) minimizes the stabilized model $f_k^S(\cdot)$, is given by

$$\delta_k^N := \delta_k^M - \frac{\alpha_k}{t_k}|u_{k+1} - \hat{u}_k|^2, \text{ for some } \alpha_k \in [0, 1]. \tag{3.6}$$

Finally, the effective decrease has the expression $\delta_k^E = f(\hat{u}_k) - f(u_{k+1})$.

We shall see in Proposition 6.1 that driving the model decrease to zero is an important ingredient for the convergence analysis. The other two measures are involved in the rule deciding when the iterate is "good enough". For u_{k+1} to become the next center, the difference $f(u_{k+1}) - f(\hat{u}_k) = -\delta_k^E$ should be sufficiently negative. The *descent test*

$$m\,\delta_k^N \leqslant \delta_k^E \quad \text{for a given parameter}\, m \in (0, 1), \tag{3.7}$$

is used by the algorithm to decide between making

either a *descent step*	or	a *null step*
(3.7) holds		(3.7) does not hold
move the center: $\hat{u}_{k+1} = u_{k+1}$		keep the current center: $\hat{u}_{k+1} = \hat{u}_k$.

3.2.2 Inexact oracles

To make the rule (3.7) precise, the set of descent parameters should be

$$\mathbf{D} = \{m, \delta_k^N\text{- by choosing}\, \alpha_k \text{ and } \delta_k^M \text{ in (3.6) and } \delta_k^E\}.$$

When the oracle output has some error neither $f(u_{k+1})$ nor $f(\hat{u}_k)$ are available: only estimates $f_{u_{k+1}}$ or $f_{\hat{u}_k}$ are at hand and some representatives for the decrease measures need to be defined. Regarding the model decrease, we let

$$\delta_k^M := \ell_k - f_k^M(u_{k+1}), \tag{3.8}$$

where the substitute for $f(\hat{u}_k)$ is a "level" ℓ_k chosen so that the inequality $\ell_k \geqslant f_{\hat{u}_k}$ holds:

$$\begin{cases} \ell_k \in [f_{\hat{u}_k}, f(\hat{u}_k)] & \text{if } \eta_u^g \equiv 0, \text{ i.e., the oracle is of lower type, and} \\ \ell_k = f_{\hat{u}_k} & \text{otherwise.} \end{cases} \tag{3.9}$$

Theorem 4.5 below shows that the level, to be specified in the set **P** , is a natural estimate for the optimal value of (1.1).

Regarding the nominal decrease (3.6), only non-negative δ_k^N in (3.7) give centers with strictly decreasing function values. We shall see in Sect. 5 that in some situations, for example with upper oracles, to ensure $\delta_k^N \geqslant 0$ some corrective action, called of *noise attenuation*, needs to be introduced.

Finally, for the effective decrease only two choices have been proposed in the literature:

- *Observed decrease* $\delta_k^E = f_{\hat{u}_k} - f_{u_{k+1}}$ as in [12,15], regardless of the oracle noise.
- *Realistic decrease* $\delta_k^E = \hat{f}_k - f_{u_{k+1}}$ as in [6,8], for the "threshold" between centers

$$\hat{f}_k := \max \left\{ \begin{matrix} f_{\hat{u}_k} \\ \max_{j \leqslant k} f_j^M(\hat{u}_k) \text{ for iterations } j \text{ following the one generating } \hat{u}_k \end{matrix} \right\}. \tag{3.10}$$

When (3.5) holds, the threshold takes a better account of reality than $f_{\hat{u}_k}$. To compute it, one sets $\hat{f}_k := f_k^M(\hat{u}_k)$ after a descent step and $\hat{f}_k = \max\{f_k^M(\hat{u}_k), \hat{f}_{k-1}\}$ after a null step.

Of these two choices, the observed one is the only possible proposal when the oracle accuracy is not controllable and the error bound is unknown. Convergence (up to the model and oracle precision) can be shown in this case using our synthetic theory; see Sect. 7.2. On the other hand, the realistic decrease is appealing but, as shown in Sect. 7.1.3, to ensure convergence the oracle must be partly asymptotically exact with errors vanishing fast enough.

For a lower oracle with a *known* error bound (such as the controllable oracle in Example 2.2) and with a lower model, there is a third type of decrease:

- *Conservative decrease* $\delta_k^E = (f_{\hat{u}_k}+\bar{\eta}_{\hat{u}_k})-(f_{u_{k+1}}+\bar{\eta}_{u_{k+1}})$, involving the knowledge of the oracle error upper bounds $\bar{\eta}_{\hat{u}_k}$ and $\bar{\eta}_{u_{k+1}}$. Thanks to this additional information, the corresponding new method, the *Controllable Bundle Algorithm* 5.4, eventually solves (1.1) up to the oracle accuracy at descent steps; see Corollary 6.12.

4 Main ingredients in the algorithm

Coping with inexact oracles increases substantially the technicalities. To clarify notation, we adopt the convention that a superscript $(\cdot)^M$ [resp. $(\cdot)^S$, resp. a hat $\widehat{(\cdot)}$] connotes an item attached to the original model [resp. the objective function in (3.3), resp. the stability center].

4.1 Aggregate objects and algorithmic pattern

Once (3.3) is solved to produce the next iterate, two key objects are the *aggregate linearization* $f_{-k}^L(\cdot)$ and *aggregate subgradient* $\hat{g}_k$ introduced below.

Lemma 4.1 (Aggregate objects) *If the model $f_k^M(\cdot)$ is convex,*

$$u_{k+1} = \hat{u}_k - t_k\hat{g}_k, \quad \textit{for some } \hat{g}_k \in \partial f_k^M(u_{k+1}) \tag{4.1}$$

is the unique solution of the master-program (3.3) and the affine function

$$f_{-k}^L(u) := f_k^M(u_{k+1}) + \langle \hat{g}_k, u - u_{k+1}\rangle \tag{4.2}$$

is an underestimate of the model: $f_{-k}^L(\cdot) \leqslant f_k^M(\cdot)$.

Proof Since its objective function is finite-valued and strongly convex, (3.3) has a unique solution characterized by the optimality condition $0 \in \partial f_k^S(u) = \partial f_k^M(u) + (u-\hat{u}_k)/t_k$, which gives (4.1). The inequality $f_{-k}^L(\cdot) \leqslant f_k^M(\cdot)$ is just the subgradient relation. □

As a consequence of (4.1) the nominal decrease in (3.8) has the equivalent expressions

$$\delta_k^N = \delta_k^M - \alpha_k t_k |\hat{g}_k|^2 = [\ell_k - f_k^M(u_{k+1})] - \alpha_k t_k |\hat{g}_k|^2 \text{ for some } \alpha_k \geqslant 0. \quad (4.3)$$

Without further specification of the sets **P** and **D**, an abstract algorithmic pattern, depending on these objects, can now be outlined.

Algorithm Pattern 4.2 (P, D) Having **P** and **D**, a starting point u_1 is chosen. The oracle output (f_{u_1}, g_{u_1}) is available. Set $k = 1$ and initialize $\hat{u}_1 = u_1$.

STEP 1. Having the model and $t_k > 0$ defined by **P**, solve (3.3) to obtain $\hat{g}_k$, u_{k+1} and $f_{-k}^L(\cdot)$ as in Lemma 4.1. Based on the definitions in **D**, compute the nominal decrease (4.3) and determine the need of *noise attenuation*.
Loop in Step 1 until noise needs no more attenuation.

STEP 2. Call the oracle at u_{k+1} to obtain the output $(f_{u_{k+1}}, g_{u_{k+1}})$.

STEP 3. With the objects in **D**, the descent test (3.7) decides between making

a *Descent step*	or	a *Null step*
(3.7) holds		(3.7) does not hold
set $\hat{u}_{k+1} = u_{k+1}$		set $\hat{u}_{k+1} = \hat{u}_k$
choose $f_{k+1}^M(\cdot) \in \mathbf{P}$		choose $f_{k+1}^M(\cdot) \in \mathbf{P}$ satisfying $f_{k+1}^M \geqslant \max\{f_{-k}^L(\cdot), f_{k+1}^L(\cdot)\}$.

STEP 4. Increase k by 1 and loop to Step 1. □

As our aim is to state general principles for convergence, this is an algorithmic *pattern*, independent of the bundle variant and its particular choices for **P** and **D**. The convergence analysis in Sect. 6 is done without specifying these sets. Section 7 reviews several instances,

$$\text{Specific Algorithm} = \text{Alg. Pattern4.2}(\mathbf{P}, \mathbf{D}) \text{ for specific } \mathbf{P}, \mathbf{D}.$$

For the new Algorithm 5.4, concrete sets **P** and **D** are given in Sect. 5.2.

Step 1 will be endowed with a stopping test based on the approximate optimality inequality (4.12) below. Also, Step 1 may also need to incorporate a loop of noise attenuation. This mechanism ensures eventual non-negativity of δ_k^N, so that the rule (3.7) resembles a descent test. We shall see in Sect. 5 how to deal with this issue by modifying Step 1 as in (5.5); if the noise attenuation loop in the reformulated Step 1 is infinite, Corollary 5.3 shows optimality of the current center (up to the model and oracle precision).

Naturally, the model cannot be totally arbitrary: Step 3 (right branch) imposes satisfaction of lower bounds. Upper bounds are also required; they are somewhat linked with the choice of t_k and will arise in Sect. 6.2, specifically in relation (6.5) therein.

4.2 Measuring optimality

To establish approximate optimality conditions it is best to refer information to the center $\hat{u}_k$.

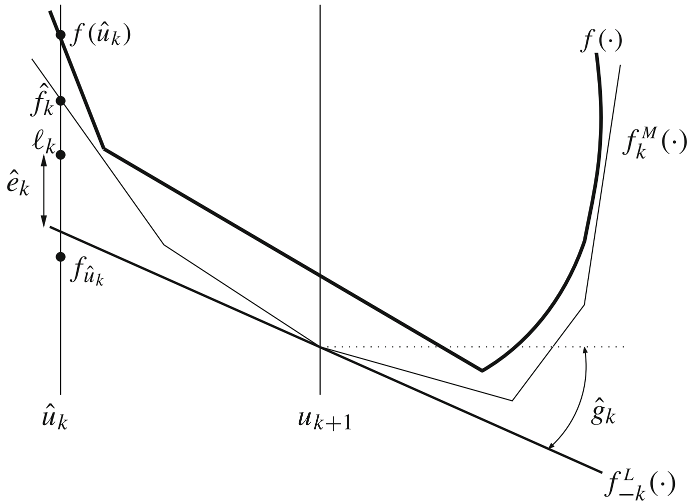

Fig. 1 Translation of the origin to the stability elements and aggregate gap $\hat{e}_k$

4.2.1 Shifting the bundle information

With exact oracles the origin is translated to the stability elements $(\hat{u}_k, f(\hat{u}_k))$. For inexact oracles, the level ℓ_k replaces $f(\hat{u}_k)$ and defines the *aggregate linearization gap*, illustrated in Fig. 1 for a lower model and oracle:

$$\hat{e}_k := \ell_k - f_{-k}^L(\hat{u}_k). \tag{4.4}$$

Combining (4.4), (4.2) and (4.1) gives for the model decrease (3.8) the expression

$$\delta_k^M = \hat{e}_k + t_k|\hat{g}_k|^2, \tag{4.5}$$

traditionally used as an optimality criterion with exact oracles. As explained next, small values for δ_k^M can be deceiving when the oracle delivers inexact information.

Remark 4.3 (Noise Attenuation) Recall from (3.9) that for lower oracles the level can be any value between $f_{\hat{u}_k}$ and $f(\hat{u}_k)$. For the function in Fig. 1, if the level is $\ell_k = f_{\hat{u}_k}$ the aggregate gap becomes negative and in (4.5) the model decrease may get close to zero. However, the figure shows that the center $\hat{u}_k$ is far from near optimality: rather than stopping, the algorithm should look for iterates further from $\hat{u}_k$. This is the basis of *noise attenuation* to cope with negative gaps, a track initiated by [12], deeply developed in [15] and described in (5.5) below: increasing t_k diminishes the attraction toward the suspect center; and (4.5) shows that δ_k^M increases (unless $\hat{g}_k$ vanishes). $\square$

The identity $f_k^M(u_{k+1}) = f_{-k}^L(\hat{u}) - \langle \hat{g}_k, \hat{u}_k - u_{k+1}\rangle$ from (4.2), gives for the aggregate linearization the equivalent expression

$$f_{-k}^L(u) = \ell_k - \hat{e}_k + \langle \hat{g}_k, u - \hat{u}_k\rangle, \tag{4.6}$$

from which follow the useful relations below, derived from (4.1):

$$\hat{e}_k = \ell_k - f_k^M(u_{k+1}) - \langle \hat{g}_k, \hat{u}_k - u_{k+1} \rangle = \ell_k - f_k^M(u_{k+1}) - t_k|\hat{g}_k|^2. \tag{4.7}$$

At this point we introduce an important convergence parameter:

$$\phi_k := \hat{e}_k + \langle \hat{g}_k, \hat{u}_k \rangle = \ell_k - f_{-k}^L(\hat{u}_k) + \langle \hat{g}_k, \hat{u}_k \rangle. \tag{4.8}$$

As shown in Theorem 4.5 below, proving convergence for an algorithm amounts to

$$\text{finding a } K^\infty - \text{subsequence } \{(\phi_k, \hat{g}_k)\} \text{ converging to } (\phi, 0) \text{ with } \phi \leqslant 0, \tag{4.9}$$

for certain infinite iteration sets K^∞ generated by the considered algorithm, so the stopping criterion in the set **P** checks that both ϕ_k and $\hat{g}_k$ are sufficiently small.

Remark 4.4 (*Duality gap interpretation of* ϕ_k) For the primal problem (2.4) in Example 2.1, (1.1) comes from Lagrangian relaxation or column generation and ϕ_k has a nice interpretation as a duality gap. Recall that the dual function is given by

$$f(u) := \max_{x \in X} L(x, u), \quad \text{with } L(x, u) := h(x) + \langle u, c(x) \rangle\,.$$

Calling x_j the primal point computed at u_j, the oracle output is

$$f_{u_j} = L(x_j, u_j) = h(x_j) + \langle u_j, c(x_j) \rangle \quad \text{and} \quad g_{u_j} = c(x_j).$$

The Lagrangian is affine with respect to u, so $f_j^L(\cdot) = L(x_j, \cdot)$. Moreover, when the model is the cutting-plane function, a subgradient $\hat{g}_k$ of the max-function $f_k^M(\cdot)$ is a convex combination of active subgradients. Hence, for $\lambda_j \geqslant 0$ such that $\sum_j \lambda_j = 1$, $\hat{g}_k = \sum_j \lambda_j g_{u_j} = \sum_j \lambda_j c(x_j)$, and for each j such that $\lambda_j > 0$, $f_k^M(u_{k+1}) = f_j^L(u_{k+1}) = L(x_j, u_{k+1}) = h(x_j) + \langle u_{k+1}, c(x_j) \rangle$. Thus,

$$\begin{aligned} f_k^M(u_{k+1}) &= \textstyle\sum_j \lambda_j h(x_j) + \left\langle u_{k+1}, \sum_j \lambda_j c(x_j) \right\rangle && \text{[by convex combination]} \\ &= \textstyle\sum_j \lambda_j h(x_j) + \langle u_{k+1}, \hat{g}_k \rangle && \text{[by the expression above for } \hat{g}_k\text{]} \\ &= \ell_k - \hat{e}_k - \langle \hat{g}_k, \hat{u}_k - u_{k+1} \rangle && \text{[by (4.7) and (4.1)]} \end{aligned}$$

Using the first identity in (4.8), we have therefore proved that

$$\sum_j \lambda_j h(x_j) = \ell_k - \hat{e}_k - \langle \hat{g}_k, \hat{u}_k \rangle = \ell_k - \phi_k. \tag{4.10}$$

Assume for simplicity that (2.4) is a linear program: $h(x) = \langle h, x \rangle$, $c(x) = Ax - a$, and introduce the primal point $\hat{x}_k := \sum_j \lambda_j x_j$. By (4.10), $\langle h, \hat{x}_k \rangle = \ell_k - \phi_k$ and, as ℓ_k from (3.9) is essentially a functional dual value,

$$\phi_k = \ell_k - \langle h, \hat{x}_k \rangle \text{ estimates the duality gap and}$$
$$\hat{g}_k = A(\hat{x}_k - a) \text{ is a constraint value.}$$

So both ϕ_k and $\hat{g}_k$ should be driven to zero (at least with exact oracles, having $\ell_k = f(\hat{u}_k)$). □

4.2.2 Convergence: what it means

The following weakened form of (3.5), extends (3.2) to a general model:

$$\text{for some } \eta^M \geqslant 0 \text{ the inequality } f_k^M(\cdot) \leqslant f(\cdot) + \eta^M \text{ holds for all } k, \tag{4.11}$$

The condition is automatic for lower oracles and models (with $\eta^M \equiv 0$) and for upper oracles, it determines the degree of agreement between the model and the true function.

Theorem 4.5 (Conditions for approximate optimality) *Suppose the model satisfies* (4.11). *If the Algorithmic Pattern 4.2 generates a K^∞-subsequence $\{(\phi_k, \hat{g}_k)\}$ satisfying* (4.9)*, then the level defined in* (3.9) *eventually estimates the infimal value of f. Namely,*

$$\limsup_{k \in K^\infty} \ell_k \leqslant \inf f(\cdot) + \phi + \eta^M \leqslant \inf f(\cdot) + \eta^M. \tag{4.12}$$

Consider an index set $K' \subset K^\infty$ such that $\{\hat{u}_k\}_{K'}$ has a limit $\hat{u}$ and define the corresponding asymptotic oracle error at descent steps

$$\eta^\infty := \liminf_{k \in K'} \eta_{\hat{u}_k}. \tag{4.13}$$

Then (4.9) *implies that $\hat{u}$ is an $(\eta^\infty + \eta^M)$-solution to problem* (1.1).

Proof Use Lemma 4.1 and (4.11) in (4.6): for all u and all k,

$$f(u) + \eta^M \geqslant f_k^M(u) \geqslant f_{-k}^L(u) = \ell_k - \phi_k + \langle \hat{g}_k, u \rangle.$$

Hence $\langle \hat{g}_k, u \rangle - f(u) \leqslant \phi_k - \ell_k + \eta^M$; take the supremum over u to obtain that

$$\sup_u \{\langle \hat{g}_k, u \rangle - f(u)\} =: f^*(\hat{g}_k) \leqslant \phi_k - \ell_k + \eta^M.$$

In view of (4.9), by closedness of the conjugate $f^*(\cdot)$,

$$f^*(0) \leqslant \liminf_{k \in K^\infty} f^*(\hat{g}_k) \leqslant \lim_{k \in K^\infty} \phi_k + \liminf_{k \in K^\infty} (-\ell_k) + \eta^M = \phi - \limsup_{k \in K^\infty} \ell_k + \eta^M,$$

which is just (4.12) since the value of the conjugate function at zero satisfies $f^*(0) = \inf f(\cdot)$. To see the final statement, given the iterate subsequence defined over $K' \subset K$, consider $k \in K'$ and use (3.9) in the first line of (1.2), written at $u = \hat{u}_k$: $f(\hat{u}_k) - \eta_{\hat{u}_k} = f_{\hat{u}_k} \leqslant \ell_k$. Passing to the limit yields the desired relation, by lower semicontinuity of $f(\cdot)$:

$$f(\hat{u}) - \eta^{\infty} \leqslant \limsup_{k \in K'} [f(\hat{u}_k) - \eta_{\hat{u}_k}] \leqslant \limsup_{k \in K'} \ell_k \leqslant \inf f(\cdot) + \eta^M.$$

□

Instead of the asymptotic condition (4.9), seemingly introduced for the first time in [15], convergence for bundle methods has always been established by

$$\text{finding a } K^{\infty} - \text{subsequence } \{(\hat{e}_k, \hat{g}_k)\} \text{ converging to } (0{,}0). \tag{4.14}$$

The difference is subtle indeed: both properties are equivalent when $\{\hat{u}_k\}$ is bounded; precisely, condition (4.9) gives an elegant argument in the unbounded case, which was overlooked before, for example in [3,13].

Theorem 4.5 gives some insight on the role played by ℓ_k, in particular on the reason for its definition (3.9). The aim of the Algorithmic Pattern 4.2 is of course to estimate as accurately as possible the optimal value and a solution of (1.1). The latter is done by means of the stability center $\hat{u}_k$ while the former can be accomplished in various ways. A straightforward approximation for the optimal value is $f_{\hat{u}_k}$, but better can be done when both the oracle and the model are of lower type (both (2.3) and (3.5) hold). In this case, the value $\hat{f}_k$ from (3.10) is more accurate than $f_{\hat{u}_k}$ and having in (4.12) that $\ell_k = \hat{f}_k$ ensures by (3.10) that the estimate is the largest available functional value. Nevertheless, notice that such definition for the level is acceptable only if $\hat{f}_k$ satisfies the relations in (3.9);

$$\hat{f}_k \in [f_{u_k}, f(u_k)] \text{ for lower oracles and } \hat{f}_k = f_{u_k} \text{ otherwise.}$$

The inequality $\hat{f}_k \geqslant f_{\hat{u}_k}$ always holds by the definition (3.10). For lower oracles and lower models, (3.5) ensures in addition that $\hat{f}_k \leqslant f(\hat{u}_k)$ and, hence, it is possible to set $\ell_k = \hat{f}_k$.

Theorem 4.5 also clarifies how a "partly asymptotically exact" lower oracle can yield exact optima in the limit, without any error in spite of the oracle inexactness. If the oracle is eventually exact only at descent steps, $\eta^{\infty} = 0$ and $f(\hat{u}_k) - f_{\hat{u}_k} \to 0$, so (3.9) implies $\ell_k - f(\hat{u}_k) \to 0$. If, in addition, the model is of lower type, (3.5) gives $\eta^M = 0$ in (4.11) and in (4.12) the only possible value for ϕ is zero (like in (4.14)).

5 On noise management and a concrete instance

We now make precise the *noise attenuation* loop in Step 1. We also provide a particular instance for the sets **P** and **D** in the Algorithmic Pattern 4.2, the Controllable Bundle Algorithm 5.4, a new method with the ability of managing the oracle accuracy.

5.1 Properties of the aggregate gap

The rewriting (4.5) suggests that a small δ_k^M might bring closer the traditional convergence property (4.14) for "not too negative" gaps. The relevance of the sign of $\hat{e}_k$

was also noticed empirically for Fig. 1 in Remark 4.3. The noise attenuation mechanism will be triggered when $\hat{e}_k$ becomes negative. Below we show how the gap sign relates with objects in the set $\mathbf{D}$, in particular with the sign of the nominal decrease δ_k^N, whose non-negativity is fundamental for (3.7) to be a genuine descent test.

Lemma 5.1 (Aggregate gap properties relevant for noise detection) *In the Algorithmic Pattern 4.2, consider the level, best function value, model and nominal decreases, and gap defined, respectively, in* (3.9), (3.10), (3.8), (4.3), *and* (4.4). *The following holds.*

(i) $\hat{e}_k \geqslant \ell_k - \hat{f}_k$ *at all iterations.*

(ii) Satisfaction of the inequality

$$\hat{e}_k \geqslant -\beta_k t_k |\hat{g}_k|^2 \text{ for some } \beta_k \in [b, 1-b] \text{ with } b \in (0, \frac{1}{2}], \tag{5.1}$$

is equivalent to any of the relations below

$$\delta_k^M \geqslant \max\left\{\hat{e}_k, (1-\beta_k) t_k |\hat{g}_k|^2\right\} \iff \delta_k^N \geqslant \Big(1 - (\alpha_k + \beta_k)\Big) t_k |\hat{g}_k|^2. \tag{5.2}$$

In particular, whenever (5.1) *holds the model decrease is non-negative. Furthermore, if we assume in* (4.3) *and* (5.1) *that*

$$\alpha_k + \beta_k \leqslant 1 - b, \tag{5.3}$$

whenever (5.1) *holds the nominal decrease is non-negative too.*

(iii) If the model satisfies (4.11) *then* $\hat{e}_k \geqslant -\big(\eta_{\hat{u}_k} + \eta^M\big)$.

Proof By definition (4.4) and the model subgradient inequality in Lemma 4.1,

$$\hat{e}_k = \ell_k - f_{-k}^L(\hat{u}_k) \geqslant \ell_k - f_k^M(\hat{u}_k). \tag{5.4}$$

The first item follows from adding $\pm \hat{f}_k$ to the right hand side, recalling the definition for $\hat{f}_k$ in (3.10). The second item follows from some simple algebra using (4.5) and (4.3).
For the third item, use the level definition (3.9) and the model assumption (4.11) in (5.4) to write $\hat{e}_k \geqslant f_{\hat{u}_k} - f_k^M(\hat{u}_k) \geqslant f_{\hat{u}_k} - f(\hat{u}_k) - \eta^M$ and (2.3) ends the proof. □

The allowed interval for β_k in (5.1), together with (5.3), implies that $\alpha_k \in [0, 1]$, so the conditions are consistent with those required for α_k in (3.6).

In view of item (ii) in Lemma 5.1, the inequality (5.1) can be used to detect when the nominal decrease is negative and a corrective action needs to be taken. This test will be incorporated in the Algorithmic Pattern 4.2 as follows.

STEP 1. Having the model and t_k defined by **P**, solve (3.3) to obtain $\hat{g}_k$, u_{k+1} and $f_{-k}^{L}(\cdot)$ as in Lemma 4.1, and compute ϕ_k from (4.8).
Stop if ϕ_k and $|\hat{g}_k|$ are both small enough. Otherwise, with the elements in **D**, compute the gap (4.4) and determine the need of *noise attenuation*:

$$\left.\begin{array}{c|c} \textit{Noisy Iteration} \quad \text{or} & \textit{Forthcoming Serious or Null Step} \\ (5.1)\text{ does not hold} & (5.1)\text{ holds} \\ \text{Keep}\begin{cases}\hat{u}_{k+1}=\hat{u}_k\\ f_{k+1}^{M}(\cdot)=f_k^{M}(\cdot)\end{cases} & \text{Go to Step 2} \\ \text{Set } t_{k+1}>t_k & \\ \text{Repeat Step 1} & \end{array}\right\} \tag{5.5}$$

When the test in Step 1 determines that noise became too large the stepsize t_k is sharply increased, inhibiting any decrease until the next descent step; see (6.14). In Corollary 5.3 we show that this simple mechanism ensures that either the current center is approximately optimal or the noise attenuation loop is finite and the algorithm proceeds to Step 2. First, we make use of the aggregate gap properties in Lemma 5.1 to examine for which models and oracles the noise attenuation test (5.1) can be dismissed.

Corollary 5.2 (Lower models and various oracles) *Suppose Algorithmic Pattern 4.2 with Step 1 from* (5.5) *uses a lower model: in the set* **P** *the model satisfies* (3.5).

- *(i) There is no need of noise attenuation and all iterations satisfy automatically* (5.1) *with β_k arbitrary, $b=0$, so $\alpha_k \in [0,1]$ in* (4.3)*, whenever one of the conditions below hold.*
  - *(ia) In the parameter set the level from* (3.9) *is given by* **P** $\ni \ell_k := \hat{f}_k$ *from* (3.10) *(recall that such a definition is possible in particular for lower oracles and models: both* (2.3) *and* (3.5) *hold).*
  - *(ib) The oracle is exact:* (2.3) *holds with $\eta_u \equiv 0$.*
- *(ii) The noise attenuation loop is finite if the oracle is lower and asymptotically exact at descent steps:*

$$(2.3)\ \textit{holds and in}\ (4.13)\ \eta_\infty = 0\ \textit{with}\ K' := \{k : \textit{the descent test (3.7) holds}\}. \tag{5.6}$$

Proof Condition (ia) implies (5.1), by Lemma 5.1(i) and (ii). When the oracle is exact, (ia) gives the result, because $\ell_k = f(\hat{u}_k)$ in (3.9) and, if the model is lower, $\hat{f}_k = f(\hat{u}_k)$ in (3.10). Similarly for (ii), reasoning asymptotically for k satisfying (3.7). □

When only finitely many descent steps are generated (K' is finite), the condition $\eta_\infty = 0$ in (5.6) in fact requires an exact evaluation of descent steps.

For general models, the noise attenuation loop can be infinite and the Algorithmic Pattern 4.2 may never reach Step 2. We use the important Theorem 4.5 to show that in this case the current center (the last descent step) is an approximate solution to (1.1).

Corollary 5.3 (Upper oracles and noisy steps) *Consider Algorithmic Pattern 4.2 with Step 1 from* (5.5) *and assume in the set* **P** *the model satisfies* (4.11).

If for some iteration $\hat{k}$ the algorithm loops forever in Step 1, then $t_k \uparrow \infty$ and the set $K^\infty := \{k \geqslant \hat{k} :$ condition (5.1) does not hold$\}$ is infinite. As a result, the last descent iterate $\hat{u} := \hat{u}_{\hat{k}}$ is an $(\eta_{\hat{u}_{\hat{k}}} + \eta^M)$-solution to (1.1).

Proof In the noise attenuation loop, t_k is increased and the stability center is maintained fixed to $\hat{u}$. The K^∞-sequence of aggregate gaps $\{\hat{e}_k\}$ is bounded below by Lemma 5.1(iii), and non-positive because (5.1) does not hold. Since the K^∞-iterates satisfy the negation of (5.1) and $\beta_k \geqslant b > 0$ therein,

$$|\hat{g}_k|^2 < -\frac{\hat{e}_k}{\beta_k t_k} \leqslant \frac{\eta_{\hat{u}_k} + \eta^M}{b\, t_k};$$

hence, $\hat{g}_k \to 0$ as t_k is driven to infinity for $k \in K^\infty$. Therefore, the limit of the K^∞-subsequence $\{\phi_k = \hat{e}_k + \langle \hat{g}_k, \hat{u}\rangle\}$ is $\phi = \lim_{K^\infty} \hat{e}_k \leqslant 0$, because (5.1) does not hold. So the convergence condition (4.9) is satisfied, Theorem 4.5 applies, and the desired result follows. □

The analysis above shows that when the Algorithmic Pattern 4.2 performs Step 1 as in (5.5), for any oracle and with a model satisfying (the very reasonable) assumption (4.11),

- either for all iterations the loop in Step 1 ends with an iterate satisfying (5.1), for which a descent or a null step will be made in Step 3;
- or for some iteration the loop in Step 1 in infinite and $\hat{u}$ is optimal up to the oracle and model precision.

For this reason the convergence analysis in Sect. 6 only considers infinite sequences stemming from the alternative in Step 3 in the Algorithmic Pattern 4.2: infinitely many descent steps or an infinite tail of consecutive null steps.

5.2 Controllable bundle method

We now state a concrete algorithm for controllable lower oracles: like in Example 2.2, high accuracy is possible yet the heavy computational burden makes it preferable to avoid exact calculations, The main novelty for this variant is in the specific choice of the level and the accuracy control of the oracle.

We assume there is an informative controllable oracle of lower type: in (1.2) $\eta_u^g \equiv 0$, so (2.3) holds. Also, the input error bound $\bar{\eta}_{u_k}$ is sent to the oracle together with the evaluation point u_k to obtain $f_{u_k} \in [f(u_k) - \bar{\eta}_{u_k}, f(u_k)]$ and an approximate subgradient g_{u_k}.

The parameter set is given by

$$\mathbf{P} = \left\{ \begin{array}{l} \text{a cutting-plane model } f_k^M(\cdot) \\ \quad = \max\left\{ f_j^L(\cdot) : j \in J_k \subseteq \{-(k-1)\} \cup \{1, \dots, k\} \right\} \\ \text{a stopping test checking if } \phi_k \text{ and } \hat{g}_k \text{ are sufficiently small;} \\ \text{the proximal stabilization } \frac{1}{2t_k}|\cdot - \hat{u}_k|^2 \text{ and an updating rule for } t_k : \\ \quad \text{- if descent step, } t_{k+1} \geqslant t_k \\ \quad \text{- if null step, } t_{k+1} = \max\{t_{\mathtt{low}}, \sigma t_k\} \text{ for } \sigma \in (0, 1] \text{ and } t_{\mathtt{low}} > 0 \\ \text{the current stability center } \hat{u}_k \text{ and its level } \ell_k := \hat{f}_k \text{ from(3.10)} \\ \text{a rule to update the oracle error bound: } \bar{\eta}_{u_{k+1}} = \bar{\eta}_{\hat{u}_k} + f_{\hat{u}_k} - \ell_k. \end{array} \right\}. \tag{5.7}$$

The model function is not the pure cutting-plane model, whose index set is $J_k = \{1, \dots, k\}$. Here, the aggregate linearization enters the index set, so the polyhedral model allows for *bundle compression*, and (3.4) is satisfied.

The descent set has the elements:

$$\mathbf{D} = \left\{ \begin{array}{l} m \in (0, 1), \alpha_k \in [0, 1] \\ \delta_k^N := \ell_k - f_k^M(u_{k+1}) - \alpha_k t_k |\hat{g}_k|^2 \\ \delta_k^E := f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - f_{u_{k+1}} - \bar{\eta}_{u_{k+1}}. \end{array} \right\}. \tag{5.8}$$

Algorithm 5.4 Controllable Bundle Method (=Alg.Pattern 4.2 with **P**,**D** from (5.7), (5.8))

The user chooses the starting point u_1 and $t_1 \geqslant t_{\mathtt{low}}$. The oracle output (f_{u_1}, g_{u_1}) is available. Set $k = 1$ and initialize $\hat{u}_1 = u_1$, $J_1 = \{1\}$, and $\ell_1 = f_{u_1}$.

Step 1. Obtain $\hat{g}_k, u_{k+1}$ and $f_{-k}^L(\cdot)$ from Lemma 4.1. by solving the quadratic program

$$\min_{r,u} r + \frac{1}{2t_k}|u - \hat{u}_k|^2 \text{ s.t. } r \geqslant f_j^L(u), j \in J_k.$$

Compute ϕ_k as in (4.8); if ϕ_k and $|\hat{g}_k|$ are small enough, stop.

Step 2. Update the oracle error bound $\bar{\eta}_{u_{k+1}} = \bar{\eta}_{\hat{u}_k} + f_{\hat{u}_k} - \ell_k$ and call the oracle with input $(u_{k+1}, \bar{\eta}_{u_{k+1}})$ to obtain the output $(f_{u_{k+1}}, g_{u_{k+1}})$.

Step 3. Check the descent test (3.7), or the equivalent relation:

$$f_{u_{k+1}} + \bar{\eta}_{u_{k+1}} \leqslant f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - m(\ell_k - f_k^M(u_{k+1}) - \alpha_k t_k |\hat{g}_k|^2)$$

and perform one of the steps below:

Descent step	or	*Null step*
The above inequality holds		The above inequality does not hold
Set $\hat{u}_{k+1} = u_{k+1}$		Set $\hat{u}_{k+1} = \hat{u}_k$
Choose $J_{k+1} \supset \{k+1\}$		Choose $J_{k+1} \supset \{k+1, -k\}$
Set $\ell_{k+1} = f^M_{k+1}(u_{k+1})$		Set $\ell_{k+1} = \max\{\ell_k, f^M_{k+1}(\hat{u}_{k+1})\}$
Choose $t_{k+1} \geqslant t_k$.		Choose $t_{k+1} \in \left[\max\left(\mathtt{t}_{\mathtt{low}}, \sigma t_k\right), t_k\right]$.

Step 4. Increase k by 1 and loop to Step 1. □

The level choice satisfies $\ell_k = \hat{f}_k \geqslant f_{\hat{u}_k}$ by (3.10). Also, note that the oracle accuracy is automatically adjusted, a useful feature if the initial bound $\bar{\eta}_{u_1}$ was taken too large. Nevertheless, the update in Step 2 of Algorithm 5.4, forces the error bound sequence $\{\bar{\eta}_{u_k}\}$ to be nonincreasing, so if the user chooses to start with exact calculations ($\bar{\eta}_{u_1} = 0$), the algorithm boils down to the classical proximal bundle method for exact oracles. By Remark 6.8 and Corollary 6.12, having nonincreasing accuracy at all iterations is crucial for proving convergence of the method. The partly asymptotically exact version in Sect. 7.1.4 drives $\eta_{\hat{u}_k} \to 0$, by taking in (5.8) the conservative decrease $\delta^E_k := f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - f_{u_{k+1}}$.

6 Convergence analysis

Our main purpose is to state a synthetic convergence theory for the Algorithmic Pattern 4.2, without stating neither the parameter set nor the descent rule. References to Algorithm 5.4 will be given throughout this section to make statements more concrete and guide the reader until the final Theorem 6.11. This general result is suitable for showing convergence of numerous bundle methods, including the inexact variants in [12,15]; see Sect. 7.

In the Algorithmic Pattern 4.2, Step 1 is given by (5.5) without stopping test. Once finiteness of the noise attenuation loop has been settled down by Corollaries 5.2 and 5.3, the K^∞-subsequences generated by the algorithm satisfy (5.1) and fit one of the following arguments:

- *Bundling*: null steps issued from the same center suitably improve the model;
- *Descent*: steps satisfying (3.7) will force the convergence property (4.9).

We start with several intermediate stages that are independent of the descent rule (3.7).

6.1 Common results for descent and null steps

The following technical Féjer-like identity is derived from writing (4.1) in the form $|u_{k+1} - u|^2 = |\hat{u}_k - t_k \hat{g}_k - u|^2$ and expanding the square:

$$|u_{k+1} - u|^2 - |\hat{u}_k - u|^2 = t_k^2 |\hat{g}_k|^2 + 2t_k \left\langle \hat{g}_k, u - \hat{u}_k \right\rangle \quad \text{for any } u \in \mathbb{R}^n. \tag{6.1}$$

It is useful to collect descent iterates in the set

$$\hat{K} := \{k \in \mathbb{N} : \text{the descent test (3.7) is satisfied}\}.$$

and consider in (4.9) the infinite sets

$$K^\infty := \begin{cases} \{k \in \mathbb{N} : k > \hat{k}\} & \text{if } \hat{K} \text{ is finite, with last element } \hat{k}, \\ \hat{K} & \text{if } \hat{K} \text{ has infinite cardinality.} \end{cases} \tag{6.2}$$

Once again, recall that Corollaries 5.2 and 5.3 already ruled out an infinite loop in Step 1 from (5.5), so the infinite sets K^∞ in (4.9) can only come from either a descent or a null step.

The general convergence property below does not require the model in **P** to satisfy any condition. However, we do assume the model satisfies (4.11), so that the corollaries from Sect. 5 apply and the algorithm K^∞-subsequences are well defined.

Proposition 6.1 *Suppose that in the set* **P** *the Algorithmic Pattern 4.2 has a bounded model and stepsizes bounded away from zero, so that both* (4.11) *and*

$$t_k \geqslant t_{\mathtt{low}} > 0, \quad \textit{for all } k \tag{6.3}$$

hold. If for any of the two index sets K^∞ from (6.2) *the model decrease eventually vanishes:*

$$\lim_{k \in K^\infty} \delta_k^M = 0,$$

the convergence property (4.9) *holds for such a set.*

Proof As Corollary 5.3 applies, (5.1) holds for $k \in K^\infty$ and (5.2) yields that

$$\lim_{k \in K^\infty} t_k |\hat{g}_k|^2 = 0 \quad \text{and} \quad \begin{cases} \lim_{k \in K^\infty} \hat{g}_k = 0 & \text{by (6.3)} \\ \lim_{k \in K^\infty} \hat{e}_k = 0 & \text{by (4.5).} \end{cases}$$

Therefore, for (4.9) to hold, we only need to show that $\phi = \lim_{K^\infty} \phi_k \leqslant 0$. When the index set is $K^\infty = \{k > \hat{k}\}$ this is direct from passing to the limit in the identity $\phi_k = \hat{e}_k + \langle \hat{g}_k, \hat{u} \rangle$ from (4.8), because $\hat{u}$ remains fixed to the last descent iterate. When the index set is $K^\infty = \hat{K}$, take two consecutive indices k_1 and k_2 in $\hat{K}$ and apply (6.1) written with $k = k_2$ (so that $\hat{u}_{k_2} = u_{k_1+1}$) to obtain the identity

$$|u_{k_2+1} - u|^2 - |u_{k_1+1} - u|^2 = t_{k_2} z_{k_2}(u), \;\; \text{for } z_k := t_k |\hat{g}_k|^2 + 2 \langle \hat{g}_k, u - \hat{u}_k \rangle .$$

The summation over k gives that $-\infty < -|u_0 - u|^2 \leqslant \sum_{k \in \hat{K}} t_k z_k$. Existence of some $\kappa > 0$ such that $z_k \leqslant -\kappa$ for all $k \in \hat{K}$ would imply $\sum_{k \in \hat{K}} t_k < +\infty$, which is impossible because of (6.3). Thus we have proved the relations

$$0 \leqslant \limsup_{k \in \hat{K}} z_k = \limsup_{k \in \hat{K}} \left(t_k |\hat{g}_k|^2 + 2 \left\langle \hat{g}_k, u \right\rangle - 2 \left\langle \hat{g}_k, \hat{u}_k \right\rangle \right) = -2 \liminf_{k \in \hat{K}} \left\langle \hat{g}_k, \hat{u}_k \right\rangle,$$

because $t_k |\hat{g}_k|^2 \to 0$. Since $\hat{e}_k \to 0$, passing to the limit in (4.8) gives $\phi = \liminf_{\hat{K}} \phi_k \leqslant 0$. □

The relation with previous results in the bundle literature can be seen as follows. By (4.6) and Lemma 4.1, $f_k^M(u) \geqslant f_{-k}^L(\hat{u}_k) + \left\langle \hat{g}_k, u - \hat{u}_k \right\rangle = \ell_k - \hat{e}_k + \left\langle \hat{g}_k, u - \hat{u}_k \right\rangle$ for all $u \in \mathbb{R}^n$. When the model satisfies (4.11), by the level definition (3.9) and (1.2), $f(u) + \eta^M \geqslant f(\hat{u}_k) - \eta_{\hat{u}_k} - \hat{e}_k + \left\langle \hat{g}_k, u - \hat{u}_k \right\rangle$ and, hence,

$$\text{(4.11) implies that } \hat{g}_k \in \partial_{\varepsilon_k} f(\hat{u}_k) \text{ for } \varepsilon_k := \hat{e}_k + \eta_{\hat{u}_k} + \eta^M \geqslant 0, \tag{6.4}$$

noting that $\varepsilon_k \geqslant 0$ by Lemma 5.1(iii).

For exact oracles, the arguments in Proposition 6.1 *do not* extend the standard proof of bundle methods. Such a proof is based on the property $\hat{g}_k \in \partial_{\hat{e}_k} f(\hat{u}_k)$ (not valid for inexact oracles), which allows a refinement of (6.1). If the speed of convergence of δ_k^M to zero can be assessed, better results are obtained: a weaker assumption on the stepsizes is possible and the full sequence $\{\hat{u}_k\}$ converges when $f(\cdot)$ has a nonempty set of minimizers. Not unexpectedly, a variant of Proposition 6.1 recovers these two results if the oracle noise is suitably controlled, via the asymptotic error at descent steps η_∞ introduced in (5.6).

Theorem 6.2 (Link with the (partly asymptotically) exact case) *Consider the Algorithmic Pattern 4.2 applied with an oracle of lower type that is asymptotically exact at descent steps, as in (5.6): $\eta^\infty = \liminf_{k \in \hat{K}} \eta_{\hat{u}_k} = 0$. Suppose that in the set* **P** *the model is lower and the stepsizes series diverges: (3.5) is satisfied and*

$$\sum_{k \in K^\infty} t_k = \infty.$$

The following holds.

(i) If $\lim_{k \in K^\infty} \delta_k^M = 0$, then $\liminf_{K^\infty} f(u_k) = \inf f(\cdot)$.

Suppose in addition that in **P** *the stepsize sequence is bounded from above:*

$$t_k \leqslant t^{\mathrm{up}}, \quad \textit{for all } k.$$

(ii) In the null-step-tail case ($K^\infty = \{k > \hat{k}\}$) the last descent step $\hat{u}_k \equiv u_{\hat{k}} =: \hat{u}$ satisfies

$$\hat{u} \textit{ minimizes } f(\cdot), \lim_{k \in K^\infty} u_k = \hat{u}, \textit{ and } \lim_{k \in K^\infty} f_{k-1}^M(u_k) = f(\hat{u}).$$

Furthermore, suppose both the oracle error and the model decrease series are convergent:

$$\sum_{k \in \hat{K}} \eta_{\hat{u}_k} < +\infty \textit{ in (5.6) } \textit{ and } \sum_{k \in \hat{K}} \delta_k^M < +\infty \textit{ in } \mathbf{D}.$$

(iii) *In the infinite-descent-step case ($K^\infty = \hat{K}$), for any limit point $\hat{u}$ of the sequence $\{\hat{u}_k\}_{k\in\hat{K}}$*

$$\hat{u} \textit{ minimizes } f(\cdot), \textit{ and the whole sequence } \{\hat{u}_k\}_{k\in\hat{K}} \textit{ converges to } \hat{u}.$$

Proof With the oracle and model assumptions Corollary 5.2(ii) applies and the sets K^∞ are well defined. To see (i), note that (5.2) yields that $\delta_k^M \geqslant (1-\beta_k)t_k|\hat{g}_k|^2 \geqslant (1-b)t_k|\hat{g}_k|^2 \geqslant 0$. Since $\delta_k^M \to 0$ by assumption, $t_k|\hat{g}_k|^2 \to 0$ for the considered subsequence. In (4.11), $\eta^M = 0$ by (3.5); together with (6.4) the inclusion $\hat{g}_k \in \partial_{\varepsilon_k} f(\hat{u}_k)$ holds with $\varepsilon_k := \hat{e}_k + \eta_{\hat{u}_k}$. Adding $\eta_{\hat{u}_k}$ to the left hand side inequality in (5.2) gives that $\varepsilon_k = \hat{e}_k + \eta_{\hat{u}_k} \leqslant \delta_k^M + \eta_{\hat{u}_k}$ and our assumption on $\eta_{\hat{u}_k}$ implies that $\varepsilon_k \to 0$. Then (i) is [5, Prop. 1.2], where $\hat{g}$ is denoted γ. To see (ii), first note that (4.1) implies that $|u_{k+1} - \hat{u}|^2 = t_k^2|\hat{g}_k|^2 \to 0$ because the stepsizes are bounded above by assumption. As the model decrease vanishes, (3.8) gives that $\lim_k \ell_k = \lim_k f_k^M(u_{k+1})$. Together with the level definition (3.9) and the oracle assumption (5.6), which forces $\eta_{\hat{u}} = 0$, we see that $f(\hat{u}) = f_{\hat{u}} \leqslant \lim_k f_k^M(u_{k+1}) \leqslant f(\hat{u})$. By lower semicontinuity, $f(\hat{u}) \leqslant \liminf_k f(u_{k+1})$ and (ii) follows. To prove (iii), observe first that $\hat{u}$ minimizes $f(\cdot)$ by (i). Then use that $\hat{g}_k \in \partial_{\varepsilon_k} f(\hat{u}_k)$ and write from (6.1)

$$\begin{aligned}|u_{k+1}-\hat{u}|^2 - |\hat{u}_k - \hat{u}|^2 &= t_k^2|\hat{g}_k|^2 + 2t_k\left\langle \hat{g}_k, \hat{u} - \hat{u}_k\right\rangle \\ &\leqslant t_k^2|\hat{g}_k|^2 + 2t_k[f(\hat{u}) - f(\hat{u}_k) + \varepsilon_k] \leqslant t_k^2|\hat{g}_k|^2 + 2t_k\varepsilon_k.\end{aligned}$$

The definition of ε_k and (4.5) yield that $t_k^2|\hat{g}_k|^2 + 2t_k\varepsilon_k = t_k(\delta_k^M - \hat{e}_k + 2\varepsilon_k) \leqslant 2t_k(\delta_k^M + \eta_{\hat{u}_k})$. For successive indices k_1 and k_2 in $\hat{K}$ summing the inequalities

$$|u_{k_2+1} - \hat{u}|^2 - |u_{k_1+1} - \hat{u}|^2 \leqslant 2t_k(\delta_k^M + \eta_{\hat{u}_k}),$$

together with the assumptions on $\{\eta_{\hat{u}_k}\}$ and $\{\delta_k^M\}$, implies that the rightmost side term forms a convergent series, so (iii) follows from [5, Prop. 1.3]. □

In view of Proposition 6.1, obtaining small δ_k^M will be our main concern in Sect. 6.3. We first state conditions ensuring this property when $\hat{K}$ in (6.2) is finite.

6.2 Null-step tail

As the stability center remains fixed throughout the present subsection, we use the notation $\hat{u} := \hat{u}_k$ and assume a weakened form of (4.11), holding only at $\hat{u}$:

$$\text{for some } \hat{\eta}^M \geqslant 0 \text{ the inequality } f_k^M(\hat{u}) \leqslant f(\hat{u}) + \hat{\eta}^M \text{ holds for all } k. \tag{6.5}$$

For the concrete Algorithm 5.4, the model is of lower type and the stronger condition (4.11) always holds with $(\hat{\eta}^M =)\eta^M = 0$.

The null-step situation, in (6.2) $K^\infty = \{k > \hat{k}\}$, just relies upon the memory effect implied by $f_{k+1}^M(\cdot) \geqslant f_{-k}^L(\cdot)$, triggered by the right branch in Step 3 of the Algorithmic

Pattern 4.2. We claim that when Step 3 systematically makes a null step, regardless of any descent test,

$$\limsup_{k>\hat{k}} \left[f_{u_k} - f^M_{k-1}(u_k) \right] \leqslant 0. \tag{6.6}$$

Two sources of errors make (1.1) difficult: one coming from the oracle and another from the model. Property (6.6) states that for null steps the model inexactness eventually vanishes.

Another important observation on the role of (6.6) refers to the nominal and effective decreases in (3.7). For simplicity, take $\alpha_k = 0$ in (4.3), so that $\delta^N_k = \delta^M_k$ from (3.8). Then

$$\delta^E_k = \ell_k - f_{u_{k+1}} = \delta^M_k + f^M_k(u_{k+1}) - f_{u_{k+1}} = \delta^N_k - \left[f_{u_{k+1}} - f^M_k(u_{k+1}) \right].$$

When the bracket becomes small, the effective and nominal decreases get close together. This is a little known point in bundle methods: a good (effective) decrease $\ell_k - f_{u_{k+1}}$ entails a more accurate model approximation $f_{u_{k+1}} - f^M_k(u_{k+1})$.

To establish (6.6) we state first a technical result linking successive optimal values of the master-program (3.3), based on arguments similar to the exact oracle case.

Lemma 6.3 *Consider the Algorithmic Pattern 4.2 and suppose that in the set* **P** *the model satisfies* (6.5) *and the stepsize is not increased at null steps:*

$$t_k \leqslant t_{k-1} \quad \textit{if at iteration } k-1 \textit{ the descent test (3.7) is not satisfied.}$$

For u_k and u_{k+1} obtained by a null step issued from the center $\hat{u}$ the following holds.

(i) $f^S_{k-1}(u_k) + \frac{1}{2t_{k-1}}|u_{k+1} - u_k|^2 \leqslant f^S_k(u_{k+1})$,

(ii) $f^S_{k-1}(u_k) + \frac{1}{2t_{k-1}}|\hat{u} - u_k|^2 \leqslant f(\hat{u}) + \hat{\eta}^M$,

(iii) $f^M_k(u_{k+1}) - f^M_{k-1}(u_k) \leqslant f^S_k(u_{k+1}) - f^S_{k-1}(u_k) + o_k$, *where we have set*

$$o_k := \frac{\langle u_{k+1} - u_k, \hat{u} - u_k \rangle}{t_k} = \frac{t_{k-1}}{t_k} \left\langle \hat{g}_{k-1}, u_{k+1} - u_k \right\rangle. \tag{6.7}$$

Proof For (i) and (ii) we refer to [15, Lemma 3.3]. To see (iii), by the definition in (3.3),

$$f^S_k(u_{k+1}) - f^M_k(u_{k+1}) = \frac{1}{2t_k}|u_{k+1} - u_k + u_k - \hat{u}|^2.$$

Develop the square and use $t_k \leqslant t_{k-1}$ in the right hand side to see that

$$\frac{1}{2t_k}|u_{k+1} - u_k|^2 - o_k + \frac{1}{2t_k}|u_k - \hat{u}|^2 \geqslant -o_k + \frac{1}{2t_{k-1}}|u_k - \hat{u}|^2.$$

The result follows, because $f^S_k(u_{k+1}) - f^M_k(u_{k+1}) \geqslant -o_k + f^S_{k-1}(u_k) - f^M_{k-1}(u_k)$. □

With an exact oracle, (6.6) becomes $f(u_k) - f^M_{k-1}(u_k) \to 0$, a known result, see for instance [5, Prop. 4.3]. Typically, $f^M_{k-1}(u_{k-1}) = f(u_{k-1})$; so we can write this as

$$[f(u_k) - f(u_{k-1})] + [f^M_{k-1}(u_{k-1}) - f^M_{k-1}(u_k)] \to 0,$$

easily proved with the Lipschitz property of $f(\cdot)$ and $f^M(\cdot)$ (Lemma 6.3 turns out to imply $u_k - u_{k-1} \to 0$, see (6.10) below). In the inexact case, the oracle f-values may behave erratically, as well as the successive models $f^M(\cdot)$. Since Lemma 6.3(i) implies a better behavior of the stabilized function $f^S(\cdot)$, item (iii) relates the model to the stabilized model.

Theorem 6.4 (Null steps) *Consider the Algorithmic Pattern 4.2 applied with an oracle having locally bounded inaccuracy:*

$$\forall\, R \geqslant 0,\ \exists\, \eta(R) \geqslant 0 \ \textit{ such that }\ |u| \leqslant R \implies \eta_u + \eta^g_u \leqslant \eta(R). \tag{6.8}$$

Suppose that in the set $\mathbf{P}$ *the model satisfies* (6.5) *and the stepsize is updated so that, whenever at iteration* $k-1$ *the descent test* (3.7) *is not satisfied,*

$$\textit{there exist positive } t_{\texttt{low}} \textit{ and } \sigma \in (0,1] \textit{ such that } t_k \in \Big[\max\Big(t_{\texttt{low}}, \sigma t_{k-1}\Big),\ t_{k-1}\Big].$$

Then the asymptotic property (6.6) *holds.*

Proof We first establish the preliminary results (6.9) and (6.10) below. The stepsize update satisfies the condition in Lemma 6.3. By item (i) therein, the sequence $\{f^S_{k-1}(u_k)\}$ is nondecreasing, hence bounded from below; say $f^S_{k-1}(u_k) \geqslant -M$ for all k. By Lemma 6.3(ii),

$$\frac{1}{2t_{k-1}}|\hat{u} - u_k|^2 \leqslant f(\hat{u}) + \hat{\eta}^M - f^S_{k-1}(u_k) \leqslant f(\hat{u}) + \hat{\eta}^M + M.$$

Using once more that stepsizes do not increase at null steps, we obtain that the sequence $\{u_k\}$ is bounded. By (2.1) and (2.2), $g_{u_k} \in \partial_{\eta_{u_k} + \eta^g_{u_k}} f(u_k)$, and the oracle assumption (6.8) implies that $\{g_{u_k}\}$ is bounded ([14, Prop. XI.4.1.2]). Our assumption on the stepsize implies in particular that (6.3) holds and, hence,

$$\text{the sequences } \{u_k\},\ \{\hat{g}_k = (\hat{u} - u_{k+1})/t_k\} \text{ and } \{g_{u_k}\} \text{ are bounded.} \tag{6.9}$$

By Lemma 6.3(ii), the monotone sequence $\{f^S_{k-1}(u_k)\}$ is bounded from above and has a limit. Together with Lemma 6.3(i) and using once again the monotonicity of stepsizes,

$$f^S_k(u_{k+1}) - f^S_{k-1}(u_k) \to 0 \quad \text{and} \quad u_{k+1} - u_k \to 0. \tag{6.10}$$

We now use these preliminary results to show (6.6). The right branch in Step 3 of the algorithmic pattern forces $f^M_k(\cdot) \geqslant f^L_k(\cdot)$ so, by (3.1), $f_{u_k} + \langle g_{u_k}, u - u_k\rangle = f^L_k(u) \leqslant f^M_k(u)$. In particular, when $u = u_{k+1}$

$$\begin{aligned} f_{u_k} - f^M_{k-1}(u_k) &= f^L_k(u_{k+1}) + \langle g_{u_k}, u_k - u_{k+1}\rangle - f^M_{k-1}(u_k) \\ &\leqslant f^M_k(u_{k+1}) + \langle g_{u_k}, u_k - u_{k+1}\rangle - f^M_{k-1}(u_k) \\ &\leqslant [f^S_k(u_{k+1}) - f^S_{k-1}(u_k)] + [\langle g_{u_k}, u_k - u_{k+1}\rangle] + o_k, \end{aligned}$$

by Lemma 6.3(iii). The results follows: by (6.9) and (6.10), the first two brackets tend to zero; and similarly for the last term, recalling our assumptions for the stepsize and (6.7). □

Remark 6.5 (On boundedness) Assumption (6.8) could be refined as follows: the oracle is bounded for any infinite sequence of null steps. We shall make use of this refinement for some concrete instances in Sect. 7 related to Examples 2.3 and 2.4. □

Remark 6.6 (*On the role of the lower bound* $t_{\mathtt{low}}$) Assumption (6.3) was only used to establish boundedness of the sequence $\{\hat{g}_k\}$. This assumption can be dropped if the model is bounded everywhere, i.e., if (4.11) holds instead of (6.5). To see this, notice that in this case (6.4) gives that $\hat{g}_k \in \partial_{\varepsilon_k} f(\hat{u})$ with $\varepsilon_k = \hat{e}_k + \eta_{\hat{u}} + \eta^M$. By local boundedness of the ε_k-subdifferential and by (6.8), we only need to show that $\{\hat{e}_k\}$ is bounded. The latter results from boundedness of $\{f^S_k(u_{k+1})\}$: plug (4.1) into the expression (4.7) of $\hat{e}_k$ to obtain

$$\hat{e}_k = \ell_k - f^M_k(u_{k+1}) - t_k|\hat{g}_k|^2 \leqslant \ell_k - f^S_k(u_{k+1}) \leqslant \ell_k + M \leqslant f(\hat{u}) + M,$$

where the last inequality follows from (3.9) recalling that $\hat{u}_k = \hat{u}$.
Finally, the assumption (6.3) can also be dropped for oracles of lower type that are partly asymptotically exact, as in Theorem 6.2. □

6.3 The role of the descent test and general convergence result

For the bundling argument, property (6.6) allows us to analyze when the model decrease eventually vanishes. Recall that this argument enters the game when in (6.2) we have $K^\infty = \{k > \hat{k}\}$—the descent test (3.7) does not hold. Since such a test depends on the effective decrease, below we give a sufficient condition for $\delta^M_k \to 0$ involving this decrease.

Proposition 6.7 (Effective decrease and bundling mechanism) *In the setting of Theorem 6.4, suppose that for the level in the set* **P** *and the effective decrease in the set* **D**

$$\limsup_{k>\hat{k}} \left[\ell_k - f_{u_{k+1}} - \delta^E_k\right] \leqslant 0\,; \tag{6.11}$$

then $\lim_{\hat{k}<k\to\infty} \delta^M_k = 0$.

Proof By Corollary 5.3 and Lemma 5.1 for the null step tail (5.1) holds and the model and nominal decreases satisfy the inequalities in (5.2) for all $k > \hat{k}$. Subtract the identity $f_k^M(u_{k+1}) = \ell_k - \delta_k^M$ from both sides of the negation of (3.7) and use (4.3) to obtain

$$-\delta_k^E - f_{u_{k+1}} + f_{u_{k+1}} - f_k^M(u_{k+1}) > -\ell_k + \delta_k^M - m\,\delta_k^N.$$

Since in (4.3) the parameter $\alpha_k \geqslant 0$, $\delta_k^N \leqslant \delta_k^M$, and reordering terms we obtain that

$$(1 - m\,)\delta_k^M < z_k + f_{u_{k+1}} - f_k^M(u_{k+1}) \text{ for } z_k := (\ell_k - f_{u_{k+1}}) - \delta_k^E.$$

By Theorem 6.4, the property (6.6) holds, together with (6.11) we obtain in the limit that

$$(1 - m\,) \limsup \delta_k^M \leqslant \limsup\big[z_k + f_{u_{k+1}} - f_k^M(u_{k+1})\big] \leqslant \limsup z_k \leqslant 0.$$

The result follows, recalling that $m \in (0, 1)$ and, by (5.2), $\delta_k^M \geqslant 0$. □

Remark 6.8 (*Interpretation for Algorithm* 5.4) Condition (6.11) helps analyzing the impact of different definitions for the effective decrease. For the *conservative* decrease given in Algorithm 5.4, (6.11) is satisfied because $\ell_k = \hat{f}_k$ and

$$\begin{aligned}
\ell_k - f_{u_{k+1}} - \delta_k^E &= \ell_k - f_{u_{k+1}} - (f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - f_{u_{k+1}} - \bar{\eta}_{u_{k+1}}) \\
&= \ell_k - f_{\hat{u}_k} - \bar{\eta}_{\hat{u}_k} + \bar{\eta}_{u_{k+1}} \\
&= 0,
\end{aligned}$$

where the last equality follows from the updating rule $\bar{\eta}_{u_{k+1}} = \bar{\eta}_{\hat{u}_k} + f_{\hat{u}_k} - \ell_k$ for the error bounds. If we were to take the same level $\ell_k = \hat{f}_k$, but use instead the *realistic* decrease $\delta_k^E = f_{\hat{u}_k} - f_{u_{k+1}}$, condition (6.11) would not hold. Indeed, reasoning like above,

$$\begin{aligned}
\ell_k - f_{u_{k+1}} - \delta_k^E &= \hat{f}_k - f_{u_{k+1}} - (f_{\hat{u}_k} - f_{u_{k+1}}) \\
&= \hat{f}_k - f_{\hat{u}_k} \\
&\geqslant 0.
\end{aligned}$$

The above inequality can be strict due to definition (3.10). In order to ensure (6.11) for this setting, the oracle should be asymptotically exact on descent steps, as in (5.6). Finally, with the *observed* decrease from [15], $\ell_k = f_{\hat{u}_k}$ and $\delta_k^E = f_{\hat{u}_k} - f_{u_{k+1}}$, condition (6.11) is satisfied for all oracles, but not necessarily (5.1)—this is straightforward from Lemma 5.1 and (6.11). For (5.1) to hold in this setting, the controllable bundle algorithm would need to incorporate the noise attenuation loop, replacing Algorithm 5.4 Step 1 by the one in (5.5). □

When the bundling argument applies, satisfaction of (6.11) is easy to accomplish (taking for example $\delta_k^E = \ell_k - f_{u_{k+1}}$). By contrast, when the algorithm generates infinitely many descent iterates ($K^\infty = \hat{K}$ in (6.2)), the working horse is the descent test (3.7). For the model decrease to vanish, the effective decrease needs to vanish too;

this is (6.12) below, a property that cannot be imposed *a priori*, but needs to be shown case by case, as in Sect. 7.

Proposition 6.9 (Effective decrease and descent mechanism) *Consider the Algorithmic Pattern 4.2 and suppose that for the sets* **P** *and* **D** *the parameters* α_k, β_k *satisfy* (5.3). *If for infinitely many iterations the descent test* (3.7) *is satisfied and*

$$\lim_{k\in\hat{K}} \delta_k^E = 0 \quad \textit{for } \hat{K} \textit{ from (6.2),} \tag{6.12}$$

then $\lim_{k\in\hat{K}} \delta_k^M = 0$.

Proof By Corollary 5.3, condition (5.1) holds and by Lemma 5.1(ii) the model and nominal decreases satisfy the inequalities in (5.2) for all $k \in \hat{K}$. In view of the assumption (5.3), both δ_k^M and $\delta_k^N \geqslant 0$ and satisfaction of (3.7) gives that

$$m\,(1-\alpha_k-\beta_k)t_k|\hat{g}_k|^2 \leqslant m\,\delta_k^N \leqslant \delta_k^E, \quad \text{for } k \in \hat{K}.$$

The result follows, because $\alpha_k+\beta_k < 1$ by (5.3) and $\delta_k^M = \delta_k^N + \alpha_k t_k|\hat{g}_k|^2$ by (4.3). □

Since with an exact oracle taking $\delta_k^E = f(\hat{u}_k) - f(u_{k+1})$ is natural, the condition

$$\text{in (1.1)} \quad \inf f(\cdot) > -\infty. \tag{6.13}$$

implies satisfaction of (6.12). We now show the same holds for Algorithm 5.4.

Remark 6.10 (*Interpretation for Algorithm* 5.4 (cont.)) Corollary 5.2(ia) Since the oracle satisfies (1.2), the effective decrease is $\delta_k^E = f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - f_{u_{k+1}} - \bar{\eta}_{u_{k+1}}$ and, hence,

$$\begin{aligned}\sum_{k\in\hat{K}} \delta_k^E &= f_{\hat{u}_1} + \bar{\eta}_{\hat{u}_1} - \lim_{k\in\hat{K}}(f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k}) \leqslant f_{\hat{u}_1} + \bar{\eta}_{\hat{u}_1} - \liminf_{k\in\hat{K}}(f(\hat{u}_k) - \eta_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k})\\ &\leqslant f_{\hat{u}_1} + \bar{\eta}_{\hat{u}_1} - [\liminf_{k\in\hat{K}} f(\hat{u}_k) + \liminf_{k\in\hat{K}}(-\eta_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k})].\end{aligned}$$

The sequence $\{-\eta_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k}\}$ is contained in $[0, \bar{\eta}_{\hat{u}_1}]$, so (6.12) follows from (6.13). □

To prove convergence of a generic proximal bundle method, oracle errors play no major role, but the stepsize needs to be updated in a manner that combines harmoniously all the requirements in the different statements. The following rule, depending on parameters $t_{\texttt{low}} > 0$ and $\sigma \in (0, 1]$, addresses this issue. The update prevents decreasing the stepsize at null steps if noise was detected via a binary variable $\texttt{na}_k$ (equal to 1 if there was noise attenuation after generating the current center $\hat{u}_k$, and zero otherwise):

$$\begin{cases} t_k \geqslant t_{k-1} + t_{\mathtt{low}} & \text{if } k-1 \text{ is a noisy step} \\ t_k \geqslant t_{\mathtt{low}} & \text{if } k-1 \text{ is a descent step} \\ t_k \in \left[\max\left(t_{\mathtt{low}}, (1-\mathtt{na}_{k-1})\sigma t_{k-1} + \mathtt{na}_{k-1} t_{k-1}\right), t_{k-1}\right] & \text{if } k-1 \text{ is a null step.} \end{cases} \tag{6.14}$$

Theorem 6.11 (Convergence) *Consider the Algorithmic Pattern 4.2 with Step 1 from* (5.5) *applied with an oracle* (1.2) *with locally bounded inaccuracy, as in* (6.8). *If*

$$\text{in the set } \mathbf{P} \begin{cases} \textit{the model satisfies (4.11),} \\ \textit{the level is given as in (3.9), and} \\ \textit{the stepsize update satisfies the rule (6.14),} \end{cases} \tag{6.15}$$

$$\text{in the set } \mathbf{D} \begin{cases} \textit{the effective decrease from (4.3) satisfies (6.11) and (6.12) and} \\ \textit{the parameters } \alpha_k \textit{ and } \beta_k \textit{ satisfy (5.3) if (5.1) is not automatic} \\ \qquad \textit{or } \alpha_k \in [0, 1] \textit{ otherwise,} \end{cases} \tag{6.16}$$

then the algorithm always generates some K^∞*-subsequence that is optimal, in the sense that* (4.9) *is satisfied and Theorem 4.5 applies.*

Proof If Step 1 needs to test (5.1) and there is an infinite loop of noise attenuation, since the update in (6.14) drives t_k to infinity in this case, Corollary 5.3 gives the result. Otherwise, the loop in Step 1 is always finite and the algorithm generates either a last descent step at iteration $\hat{k}$ followed by a null-step tail, or $\hat{K}$ in (6.2) is infinite. In the first case (6.14) satisfies the stepsize conditions in Theorem 6.4. As (3.5) implies satisfaction of (6.5) with $\hat{\eta} = \eta^M$ the theorem applies. By Proposition 6.7, the model decrease vanishes and the assertion follows from Proposition 6.1, written with $K^\infty = \{k > \hat{k}\}$. Finally, if infinitely many descent steps are generated, the assumption that $\lim_{k \in \hat{K}} \delta_k^M = 0$ follows from Proposition 6.9 and the result follows once again from Proposition 6.1, this time written with $K^\infty = \hat{K}$. □

Corollary 6.12 (Interpretation for Algorithm 5.4 (end)) *If problem* (1.1) *satisfies* (6.13)*, any limit point* $\hat{u}$ *of the Controllable Bundle Algorithm 5.4 is* η^∞*-optimal with* $\eta^\infty \leqslant \bar{\eta}_{\hat{u}_1}$.

Proof The sets **P** and **D** are given in (5.7) and (5.8), respectively. The sequence of error bounds is nonincreasing and there is no need of noise attenuation, by Corollary 5.2(ia). The update (5.7) fits the rule (6.14) and the oracle inaccuracy is controllable with a nonincreasing sequence of error bounds, so (6.8) is satisfied with $\eta(R) = \bar{\eta}_{\hat{u}_1}$. Finally, as explained in Remarks 6.8 and 6.10, both (6.11) and (6.12) are satisfied for the choices for δ_k^N and δ_k^E in (5.8). Theorem 6.11 applies with $\eta^M = 0$ because both the oracle and the model are lower. □

The final section reviews a number of methods fitting our convergence framework.

7 Instances

For each bundle variant in this section convergence follows from Theorem 6.11, by analyzing

- if the oracle is bounded, in the sense of (6.8) or the Remark 6.5; and
- if conditions (6.15) and (6.16) hold for the elements in the specific sets **P** and **D**.

Regarding the set **P**, in (6.15) the stepsize is given by (6.14), so we only need to check that the level and the model respectively satisfy (3.9) and (4.11).
As for the set **D** and (6.16), we first determine if the considered variant needs to attenuate noise. If such is the case, parameters α_k and β_k will be given by (5.3); otherwise $\alpha_k \in [0, 1]$. Here we only need to check that the effective decrease satisfies conditions (6.11) and (6.12). For this last property, we assume that problem (1.1) satisfies (6.13).

7.1 A collection of bundle methods for lower oracles and models

We review methods for lower oracles (2.3) and lower models, so that (3.5) holds and in (4.11) the error is null: $\eta^M = 0$. To keep the master-program size controlled, the cutting-plane model can be endowed with bundle compression, like in Sect. 5.2. This mechanism replaces past linearizations by the aggregate one, $f_{-k}^L(\cdot)$, so that (3.4) holds and, hence, in (3.9) taking $\ell_k = \hat{f}_k$ is possible. For all these variants, the set **P** satisfies (6.15) and Step 1 never needs to attenuate noise because (5.1) always holds.

7.1.1 Exact oracles

Both the *classical* bundle methods and the *spectral* algorithms [11,18] use an exact oracle and, hence, $\ell_k = \hat{f}_k = f(\hat{u}_k)$. The effective decrease $\delta_k^E = f(\hat{u}_k) - f(u_{k+1})$ gives (6.11), because the left hand side therein is null, as for (6.12), it is direct from (6.13). The descent test (3.7)

$$f(u_{k+1}) \leqslant f(\hat{u}_k) - m\delta_k^N \quad \text{for} \quad \delta_k^N = f(\hat{u}_k) - f_k^M(u_{k+1}) - \alpha_k t_k |\hat{g}_k|^2$$

usually takes $\alpha_k \equiv 0$ in (4.3), but any value in [0, 1] could be used instead.

7.1.2 Partially inexact oracles

The *partially inexact* proximal bundle method was introduced in [9] and revisited in [16], for a level variant see [4]. To ensure that the function information is exact at descent steps the oracle should be a particular case of the partly asymptotically exact one given in Example 2.2, with $\bar{\eta}_u = 0$ whenever $f_u \leqslant \gamma_u$. Then $\ell_k = \hat{f}_k = f(\hat{u}_k)$ and the observed decrease $\delta_k^E := f(\hat{u}_k) - f_{u_{k+1}}$ yields both a null left hand side in (6.11) and satisfaction of (6.12), by (6.13).

7.1.3 Incremental bundle method

The *incremental bundle* method [6] was developed for lower oracles with descent errors vanishing fast enough, as in (5.6). The realistic decrease $\delta_k^E = \hat{f}_k - f_{u_{k+1}}$ yields in (6.11) a null left hand side. Condition (6.12) is satisfied because $\hat{f}_k \leqslant f_{\hat{u}_k} + \eta_{\hat{u}_k}$ and, hence,

$$(0 \leqslant) \sum_{k \in \hat{K}} \delta_k^E \leqslant \sum_{k \in \hat{K}} (f_{\hat{u}_k} + \eta_{\hat{u}_k} - f_{\hat{u}_{k+1}}) = \sum_{k \in \hat{K}} (f_{\hat{u}_k} - f_{\hat{u}_{k+1}}) + \sum_{k \in \hat{K}} \eta_{\hat{u}_k} < +\infty.$$

When $\hat{K}$ is finite, the last descent step $\hat{u}$ is η^∞-solution with $\eta^\infty = \eta_{\hat{u}}$ not necessarily zero. When $\hat{K}$ is infinite, $\eta^\infty = 0$ and the limit point solves problem (1.1) exactly. The criterion (4.8) makes superfluous the *unboundedness detection* loop in [6] (see the errata in [5]).

7.1.4 Asymptotically exact bundle method

This is the proximal version of the level bundle method for oracles with on-demand accuracy [4]. The variant is suitable for lower oracles that are eventually exact at descent iterates, like the partly asymptotically exact oracle in Example 2.2: $\eta_u^g \equiv 0$ and the known error bound $\bar{\eta}_{\hat{u}_k}$ asymptotically vanishes. The conservative decrease $\delta_k^E = f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - f_{u_{k+1}}$ yields (6.11):

$$\ell_k - f_{u_{k+1}} - \delta_k^E = \hat{f}_k - (f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k}) \leqslant \hat{f}_k - f(\hat{u}_k)$$

because the oracle is lower. As for (6.12), by (6.13) combined with (1.2)

$$\begin{aligned} 0 \leqslant \sum_{k \in \hat{K}} (f_{\hat{u}_k} - f_{u_{k+1}}) &= \sum_{k \in \hat{K}} (f_{\hat{u}_k} - f_{\hat{u}_{k+1}}) \\ &= f_{\hat{u}_1} - \lim_{k \in \hat{K}} f_{\hat{u}_k} = f_{\hat{u}_1} - \lim_{k \in \hat{K}} (f(\hat{u}_k) - \eta_{\hat{u}_k}) < +\infty. \quad (7.1) \end{aligned}$$

So $(f_{\hat{u}_k} - f_{u_{k+1}}) \to 0$ and (6.12) holds because $\bar{\eta}_{\hat{u}_k} \to 0$ by the oracle assumption. The descent test (3.7) is $f_{u_{k+1}} \leqslant f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - m\delta_k^N$ for $\delta_k^N = \hat{f}_k - f_k^M(u_{k+1}) - \alpha_k t_k |\hat{g}_k|^2$.

In this method, even though the oracle never delivers exact evaluations, convergence is still exact, as in Sect. 7.1.2. To see this, it suffices to show that $\eta_{\hat{u}} = 0$ if $K^\infty = \{k > \hat{k}\}$ (when $K^\infty = \hat{K}$, $0 \leqslant \eta_{u_k} \leqslant \bar{\eta}_{u_k} \to 0$ by assumption). When there is a last descent step $\hat{u}$, the definitions of ℓ_k and δ_k^E imply that $f_{u_{k+1}} > f_{\hat{u}_k} + \bar{\eta}_{\hat{u}_k} - m\,\delta_k^N$ if (3.7) does not hold. Then

$$\begin{aligned} f_{u_{k+1}} - f_k^M(u_{k+1}) &> f_{\hat{u}} + \bar{\eta}_{\hat{u}} - m\,\delta_k^N - f_k^M(u_{k+1}) && [\text{by “not}'' \text{ (3.7)}] \\ &= f_{\hat{u}} + \bar{\eta}_{\hat{u}} - m\,(\delta_k^M - \alpha_k t_k |\hat{g}_k|^2) - f_k^M(u_{k+1}) && [\text{by (4.3)}] \\ &= f_{\hat{u}} + \bar{\eta}_{\hat{u}} - \ell_k - m\,(\delta_k^M - \alpha_k t_k |\hat{g}_k|^2) + \delta_k^M && [\text{by (3.8)}] \\ &= \big(f_{\hat{u}} + \bar{\eta}_{\hat{u}} - \hat{f}_k\big) + (1-m)\delta_k^M + m\,\alpha_k t_k |\hat{g}_k|^2. && [\text{because } \ell_k = \hat{f}_k] \end{aligned}$$

In view of (6.6), $\limsup_k \left(f_{\hat u} + \bar\eta_{\hat u} - \hat f_k\right) + (1-m)\delta_k^M + m\,\alpha_k t_k |\hat g_k|^2 \leqslant 0$. However, the second and third terms above are non-negative ($\delta_k^M \geqslant 0$ by Proposition 6.9). Similarly for the first term because, by the first line in (1.2) together with (3.10) and (3.4) evaluated at $\hat u$,

$$f_{\hat u} + \bar\eta_{\hat u} \geqslant f(\hat u) \geqslant \hat f_k. \tag{7.2}$$

Therefore $f_{u_{k+1}} - f_k^M(u_{k+1}) \to 0$, hence $\hat f_k \to f_{\hat u} + \bar\eta_{\hat u}$. In the limit, the right hand side in (7.2) yields that $f_{\hat u} + \bar\eta_{\hat u} = f(\hat u)$ and, by the first line in (1.2), $f(\hat u) \leqslant f_{\hat u} + \eta_{\hat u} \leqslant f_{\hat u} + \bar\eta_{\hat u} = f(\hat u)$, so $\eta_{\hat u} = \bar\eta_{\hat u} = 0$, as stated.

Finally note that for oracles with errors vanishing sufficiently fast, as in (5.6), if the update (6.14) takes stepsizes $t_k \leqslant t^{\mathrm{up}}$, the stronger convergence result in Theorem 6.2 holds.

7.2 Inexact bundle methods

The methods in [12,15] also fit our convergence framework. They deal with uncontrollable oracles, bounded in the sense of (6.8) or Remark 6.5, and possibly of upper type: the level (3.9) is $\ell_k = f_{\hat u_k}$. Then with any model of the form (3.4) condition (4.11) holds and the set $\mathbf{P}$ satisfies (6.15). The set $\mathbf{D}$ satisfies (6.16) with the observed decrease $\delta_k^E = f_{\hat u_k} - f_{u_{k+1}}$, because (6.11) holds and (6.12) follows directly from (7.1) (the oracle error is bounded).

Theorem 6.11 ensures convergence of the method subsequences satisfying (5.1) (an infinite null tail or infinitely many descent iterates). Corollary 5.3 gives the result for an infinite loop of noise attenuation (after a last descent step (5.1) never holds). In all cases the algorithm determines asymptotically an $(\eta_\infty + \eta^M)$-solution to problem (1.1). The present work generalizes [15] from taking $\alpha_k \equiv 0$ and $\beta_k \equiv \frac{1}{2}$ to any pair $\alpha_k + \beta_k \leqslant 1 - b$, and $\beta_k \in [b, 1-b]$ where $b \in (0, \frac{1}{2}]$.

7.3 Linearly constrained problems

All our results hold when constraining (1.1) to a nonempty polyhedron U:

$$\min f(u) \text{ s.t. } u \in U. \tag{7.3}$$

because this simple set can be introduced directly in the master-program (3.3). In Lemma 4.1, the update (4.1) adds a normal element $\nu_k \in N_U(u_{k+1})$ to the aggregate subgradient, so the aggregate linearization (4.2) is an underestimate of the function over the set U only, [15, § 2].

The crucial boundedness property (6.9) in Theorem 6.4, which now needs to hold for the sequence $\{\hat g_k + \nu_k = (\hat u - u_{k+1})/t_k\}$, still follows from the condition (6.3); we refer to [15, Lemma 3.3] for details.

7.4 Nonlinearly constrained problems

In the following generalization of Example 2.3, the convex problem

$$\min\ h(u)\ \text{ s.t. }\ u \in U \text{ and } c(u) \leqslant 0, \tag{7.4}$$

has an "easy" objective function $h(\cdot)$. a scalar constraint $c(\cdot)$ hard to evaluate and a simple polyhedral set U. Supposing a Slater condition holds for this problem, we now consider two solution methods whose convergence can be derived from our theory.

7.4.1 Using improvement functions

Letting $\bar{h}$ denote the optimal value, solving the (7.4) is equivalent to solve (7.3) with

$$f(u) := \max\{h(u) - \bar{h}, c(u)\} \quad \text{over the set } U.$$

When $\bar{h}$ is unknown, a possible replacement is the objective value at the current center, penalizing infeasibility, to prevent zigzagging. For example, the approximation in [21]:

$$f_{u_{k+1}} := \max\{h(u) - h(\hat{u}_k) - \rho_k \max(c_{\hat{u}_k}, 0), c_u - \sigma_k \max(c_{\hat{u}_k}, 0)\},$$

for parameters ρ_k, σ_k given. The notation makes explicit that the h- and c-oracles are respectively exact and inexact: the subgradient is either the exact $g^h(u_{k+1})$ or the inexact $g^c_{u_{k+1}}$, depending on which term realizes the maximum. Since the c-oracle is of upper type, so is the f-oracle and the level (3.9) is $\ell_k = f_{\hat{u}_k} = (1 - \sigma_k) \max(c_{\hat{u}_k}, 0)$. The method needs noise attenuation in Step 1, with parameters in (5.3) related to the counterparts in [21] as follows

$$2\alpha_k = \alpha_k^{vAS} \quad \text{and} \quad 2\beta_k = 1 - \beta_k^{vAS} \quad \text{for } (\alpha_k^{vAS}, \beta_k^{vAS}) \text{ from [1]}.$$

The cutting-plane models for the objective and constraint functions give the model:

$$f_k^M(u) = \max\{\check{h}_k(u) - h(\hat{u}_k) - \rho_k \max(c_{\hat{u}_k}, 0), \check{c}_k(u) - \sigma_k \max(c_{\hat{u}_k}, 0)\}.$$

Satisfaction of (4.11) follows the condition holding for $\check{c}_k(\cdot)$ because U is compact, while satisfaction of (6.8) follows from Remark 6.5.

With the (observed) effective decrease $\delta_k^E := \ell_k - f_{u_{k+1}}$, (6.11) is automatic.

Finally, to show (6.12), we consider one particular case for the penalty (the setting in [21] is more general):

$$\rho_{k+1} = \rho_k + 1 \text{ at each iteration } k \text{ satisfying (3.7) for which } c_{u_{k+1}} > 0.$$

Descent iterates are either always unfeasible or remain feasible after a first feasible center is found ($f_{u_{k+1}}$ is a maximum). In the first case, $c_{\hat{u}_k} > 0$ for all $k \in \hat{K} = K^{\infty}$.

By the penalty update, in the max-operation defining f_u the second term eventually prevails. Hence,

$$\delta_k^E = \ell_k - f_{u_{k+1}} = (1-\sigma_k)c_{\hat{u}_k} - (c_{\hat{u}_{k+1}} - \sigma_k c_{\hat{u}_k})$$
$$= c_{\hat{u}_k} - c_{\hat{u}_{k+1}} \text{ for } k \in \hat{K} \text{ sufficiently large.}$$

Summing over $k \in \hat{K}$ shows that the effective decrease series is convergent, so (6.12) holds (regardless of the value of σ_k). In the second case, descent steps are eventually feasible and

$$\delta_k^E = \ell_k - f_{u_{k+1}} = 0 - \max\{h(\hat{u}_{k+1}) - h(\hat{u}_k), c_{\hat{u}_{k+1}}\} \leqslant -h(\hat{u}_{k+1}) + h(\hat{u}_k)$$

for large $k \in \hat{K}$. Assuming once more (6.13) and taking the sum gives (6.12).

By Theorem 6.11, the algorithm limit points solve (7.3) within the accuracy $\eta^\infty + \eta^M$, and with the Slater assumption, solving this problem is equivalent to solving (7.4), as desired.

7.4.2 *Using exact penalties*

The Slater condition ensures that in (7.4) the set of Lagrange multipliers is nonempty and bounded. Therefore, for any ρ greater than the largest Lagrange multiplier, solutions to (7.4) can be found by minimizing over the set U the exact penalty function

$$f(u) := h(u) + \rho \max\{c(u), 0\},$$

i.e., the setting (7.3). As in Sect. 7.4.1, the transformation of the constrained problem into a linearly constrained one introduces an *exogenous* inaccuracy that can be easily handled together with the genuine errors in the h- and c-oracles.
With a model of the form $f_k^M(u) := h_k^M(u) + \rho \max\{c_k^M(u), 0\}$, (4.11) and (6.8) follow from the h—and c-models and oracles.
The approach bottleneck is estimating the penalty: making successive approximations of this parameter amounts to having a more accurate f-oracle. A possible penalty update is $\rho_k = \max\{\rho_{k-1}, \lambda_k + 1\}$ for a Lagrange multiplier $\lambda_k \geqslant 0$ of

$$\min_{u \in U} h_k^M(u) + \frac{1}{2t_k}|u - \hat{u}_k|^2 \text{ s.t. } c_k^M(u) \leqslant 0, \tag{7.5}$$

a Successive Quadratic Programming-like problem that gives the next iterate u_{k+1}. This is a feasible problem (by the Slater assumption) whose solution also solves

$$\min_{u \in U} h_k^M(u) + \rho_k \max\{c_k^M(u), 0\} + \frac{1}{2t_k}|u - \hat{u}_k|^2.$$

for sufficiently large ρ_k. For the approach to behave as an exact penalty method, the penalty needs to stabilize eventually. This requires the Lagrange multiplier sequence to be bounded, a property that we prove under appropriate assumptions.

Lemma 7.1 *Suppose U is a bounded set and there exists $u^0 \in U$ such that $c(u^0) < 0$. If the stepsizes satisfy (6.3) and the c-model satisfies*

$$c_k^M(\cdot) \leqslant \eta^M \quad \textit{for some bound } 0 \leqslant \eta^M < -c(u^0),$$

the sequence of Lagrange multipliers $\{\lambda_k\}$ of (7.5) is bounded.

Proof The optimality condition for (7.5) provides $p_k^h \in \partial h_k^M(u_{k+1})$, $p_k^c \in \partial c_k^M(u_{k+1})$, $p_k^u \in N_U(u_{k+1})$, and $\lambda_k \geqslant 0$ satisfying $p_k^h + \frac{u_{k+1}-\hat{u}}{t_k} + \lambda_k p_k^c + p_k^u = 0$. Without loss of generality, suppose $\lambda_k > 0$, so that $p_k = p_k^c + p_k^u/\lambda_k$ and $\lambda_k p_k = -(p_k^h + \frac{u_{k+1}-\hat{u}}{t_k})$ yield the identity $\lambda_k |p_k|^2 = -\left\langle p_k, (p_k^h + \frac{u_{k+1}-\hat{u}}{t_k})\right\rangle$. Assume for the moment $p_k \neq 0$; by Cauchy–Schwarz,

$$\lambda_k = -\frac{1}{|p_k|^2}\left(\left\langle p_k, p_k^h\right\rangle + \left\langle p_k, \frac{u_{k+1}-\hat{u}_k}{t_k}\right\rangle\right) \leqslant \frac{|p_k|}{|p_k|^2}\left(|p_k^h| + \frac{|u_{k+1}-\hat{u}_k|}{t_k}\right).$$

As U is bounded and $h_k^M(\cdot)$ is convex, then $p_k^h \in \partial h_k^M(u_{k+1})$ is bounded as well. Together with (6.3) we see that there exists a constant $M > 0$ such that $\lambda_k \leqslant M/|p_k|$ for all k.

It remains to show that the sequence $\{|p_k|\}$ is bounded away from zero. The definitions of p_k^c and p_k^u imply that $c_k^M(u_{k+1}) + \left\langle p_k^c, u^0 - u_{k+1}\right\rangle \leqslant c_k^M(u^0)$ and $\frac{1}{\lambda_k}\left\langle p_k^u, u^0 - u_{k+1}\right\rangle \leqslant 0$. By adding these two inequalities and remembering that $c_k^M(u_{k+1}) = 0$ because $\lambda_k > 0$, we get $\left\langle p_k, u^0 - u_{k+1}\right\rangle \leqslant c_k^M(u^0)$. Therefore,

$$-|p_k||u^0 - u_{k+1}| \leqslant \left\langle p_k, u^0 - u_{k+1}\right\rangle \leqslant c_k^M(u^0) \leqslant c(u^0) + \eta^M < 0,$$

where the last inequality follows from the assumption on u^0 and η^M. Since U is a bounded set, we conclude that $\liminf_k |p_k| > 0$, and hence $\{\lambda_k\}$ is a bounded sequence. □

Once the penalty parameter eventually stabilizes at a value ρ, Theorem 6.11 applies: the limit points of the sequence $\{\hat{u}_k\}$ solve the constrained problem within an accuracy bound depending also on the value $\limsup_{k\in\hat{K}}\{0, \rho - \rho_{\hat{k}}\}$, the asymptotic error made when estimating the penalty at descent steps.

7.5 Composite functions

The composite bundle method [19] uses the approximation $F(\cdot; \hat{u})$ in Example 2.4 (with $\hat{u} = \hat{u}_k$) as an economic intermediate model for the function $f(\cdot) = (h \circ c)(\cdot)$. The reason is that evaluating the f-subgradients is expensive, because they need computing the c-Jacobian:

$$\partial f(u) = Dc(u)^\top \partial h(C) \quad \text{for} \quad C = c(u).$$

To interpret this method in our setting, consider as oracle output

$$\begin{aligned} f_{u_{k+1}} &:= F(u_{k+1}; \hat{u}_k) = h(C_{k+1}) \quad \text{for} \quad C_{k+1} := c(\hat{u}_k) + Dc(\hat{u}_k)(u_{k+1} - \hat{u}_k) \\ g_{u_{k+1}} &:= Dc(\hat{u}_k)^\top G_{k+1} \quad \text{for} \quad G_{k+1} \in \partial h(C_{k+1}). \end{aligned}$$

The h-oracle is exact everywhere but the f-oracle is exact at each center $\hat{u}_k$. By smoothness of the operator c, this oracle satisfies (6.8) for each fixed $\hat{u}_k$, as in Remark 6.5.

By convexity, $\check{h}(\cdot) \leqslant h(\cdot)$ and by positive homogeneity, the model

$$f_k^M(\cdot) := \check{h}(c(\hat{u}_k) + Dc(\hat{u}_k)(\cdot - \hat{u}_k))$$

stays below $F(\cdot; \hat{u}_k)$, but not necessarily below $f(\cdot)$. An interesting feature of the approach is that, even though the model is not of lower type ((3.5) may not hold), the method *does not need* to attenuate noise, thanks to the special model structure. More precisely, first note that

$$f_k^M(\hat{u}_k) = (\check{h} \circ c)(\hat{u}_k)) \leqslant (h \circ c)(\hat{u}_k)) = f(\hat{u}_k).$$

Then, because $\hat{g}_k \in \partial f_k^M(u_{k+1})$ by Lemma 4.1, from (4.6) we see that

$$f(\hat{u}_k) = f_k^M(\hat{u}_k) \geqslant f_k^M(u_{k+1}) + \langle \hat{g}_k, \hat{u}_k - u_{k+1} \rangle = f_{-k}^L(\hat{u}_k).$$

So taking as level $\ell_k = f(\hat{u}_k)$ gives a null aggregate gap (4.4) and, by item (ii) in Lemma 5.1, there is no need of noise attenuation. In [19] a null step is declared whenever there is no descent for the approximating function, corresponding to the observed decrease

$$\delta_k^E := F(\hat{u}_k; \hat{u}_k) - F(u_{k+1}; \hat{u}_k) = f(\hat{u}_k) - h(C_{k+1}) = \ell_k - f_k^M(u_{k+1}),$$

which trivially ensures (6.11). Regarding (6.12), the composite bundle method checks a *backtracking* test before declaring a descent step. More precisely, for $k \in \hat{K}$

$$\begin{aligned} f(\hat{u}_k) - f(\hat{u}_{k+1}) &= [f(\hat{u}_k) - h(C_{k+1})] + [h(C_{k+1}) - f(\hat{u}_{k+1})] \\ &= \delta_k^E + [h(C_{k+1}) - f(\hat{u}_{k+1})]. \end{aligned}$$

The usual argument invoking (6.13) would give (6.12) if the second bracket eventually vanished. The backtracking test declares a descent step if, in addition to (3.7), the condition

$$\langle G_{k+1}, C_{k+1} - c(u_{k+1}) \rangle \geqslant -m_2 \delta_k^N \text{ for } G_{k+1} \in \partial h(c(u_{k+1}))$$

holds. Otherwise the stepsize is decreased (“backtracking”), with the same model and center. As the number of backtracking steps is finite ([19]), eventually the algorithm generates sets K^∞ as in (6.2). The backtracking condition is easy to test, because the additional oracle call does not involve the expensive Jacobian at u_{k+1}. Such a

computation is done only if the backtracking test is passed, to define the next cheap oracle, i.e. $F(\cdot; \hat{u}_{k+1})$.
Since positively homogeneous functions are support functions,

$$h(c(u_{k+1})) = \langle G_{k+1}, c(u_{k+1})\rangle \quad \text{and} \quad h(C_{k+1}) \geqslant \langle G_{k+1}, C_{k+1}\rangle\,,$$

so checking the need of backtracking is equivalent to testing if $[h(C_{k+1}) - f(\hat{u}_{k+1})] \geqslant -m_2\delta_k^N$. Together with (3.7) this means that when $K^\infty = \hat{K}$ we have that

$$f(\hat{u}_k) - f(\hat{u}_{k+1}) = \delta_k^E + [h(C_{k+1}) - f(\hat{u}_{k+1})] \geqslant (m_1 - m_2)\delta_k^N.$$

Since $m_1 - m_2 > 0$, taking the sum over $k \in K^\infty$ implies that the series of nominal decreases converges, by (6.13). So the effective decrease series converges too and (6.12) follows.

By Theorem 6.11, the limit points of the sequence satisfy (4.12), and in view of the level definition, they solve (1.1) exactly.

Acknowledgments Research partly done during a postdoctoral visit of the first author at Inria. The authors are grateful to the reviewers for many insightful comments.

References

1. BenAmor, M.T., Desrosiers, J., Frangioni, A.: On the choice of explicit stabilizing terms in column generation. Discret. Appl. Math. **157**(6), 1167–1184 (2009)
2. Cheney, E., Goldstein, A.: Newton's method for convex programming and Tchebycheff approximations. Numer. Math. **1**, 253–268 (1959)
3. Correa, R., Lemaréchal, C.: Convergence of some algorithms for convex minimization. Math. Program. **62**(2), 261–275 (1993)
4. de Oliveira, W., Sagastizábal, C.: Level bundle methods for oracles with on-demand accuracy. Optim. Methods Softw. **29**(6), 1180–1209 (2014). doi:10.1080/10556788.2013.871282
5. Émiel, G., Sagastizábal, C.: Incremental-like bundle methods with application to energy planning (corrigendum). Optimization Online Report 2147 (2009)
6. Émiel, G., Sagastizábal, C.: Incremental-like bundle methods with application to energy planning. Comput. Opt. Appl. **46**(2), 305–332 (2010)
7. Frangioni, A.: Generalized bundle methods. SIAM J. Optim. **13**(1), 117–156 (2003)
8. Frangioni, A.: Private communication (2008)
9. Gaudioso, M., Giallombardo, G., Miglionico, G.: An incremental method for solving convex finite min–max problems. Math. Oper. Res. **31**(1), 173–187 (2006)
10. Goffin, J.-L., Haurie, A., Vial, J-Ph: Decomposition and nondifferentiable optimization with the projective algorithm. Manag. Sci. **38**(2), 284–302 (1992)
11. Helmberg, C., Rendl, F.: A spectral bundle method for semidefinite programming. SIAM J. Optim. **10**(3), 673–696 (2000)
12. Hintermüller, M.: A proximal bundle method based on approximate subgradients. Comput. Optim. Appl. **20**(3), 245–266 (2001). doi:10.1023/A:1011259017643
13. Hiriart-Urruty, J.-B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Springer, Heidelberg (1993). (Two volumes)
14. Kelley, J.E.: The cutting plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8**, 703–712 (1960)
15. Kiwiel, K.C.: A proximal bundle method with approximate subgradient linearizations. SIAM J. Optim. **16**(4), 1007–1023 (2006)
16. Kiwiel, K.C.: Bundle Methods for Convex Minimization with Partially Inexact Oracles. Technical Report, Systems Research Institute, Polish Academy of Sciences (April 2010)

17. Marsten, R.E., Hogan, W.W., Blankenship, J.W.: The boxstep method for large-scale optimization. Oper. Res. **23**(3), 389–405 (1975)
18. Oustry, F.: A second-order bundle method to minimize the maximum eigenvalue function. Math. Program. **89**(1), 1–33 (2000)
19. Sagastizábal, C.: Composite proximal bundle method. Math. Program. **140**(1), 189–233 (2013)
20. Solodov, M.V.: On approximations with finite precision in bundle methods for nonsmooth optimization. J. Opt. Theory Appl. **119**(1), 151–165 (2003)
21. van Ackooij, W., Sagastizábal, C.: Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. SIAM J. Optim. **24**(2), 733–765 (2014)