SIAM J. OPTIM.
Vol. 28, No. 1, pp. 379–410

INCREMENTAL BUNDLE METHODS USING UPPER MODELS[*]

WIM VAN ACKOOIJ[†] AND ANTONIO FRANGIONI[‡]

Abstract. We propose a family of proximal bundle methods for minimizing sum-structured convex nondifferentiable functions which require two slightly uncommon assumptions that are satisfied in many relevant applications: Lipschitz continuity of the functions and oracles which also produce *upper estimates* on the function values. In exchange, the methods: (i) use *upper models* of the functions that allow one to estimate function values at points where the oracle has not been called; (ii) provide the oracles with more information about when the function computation can be interrupted, possibly diminishing their cost; (iii) allow one to skip oracle calls entirely for some of the component functions, not only at "null steps" but also at "serious steps"; (iv) provide explicit and reliable a posteriori estimates of the quality of the obtained solutions; (v) work with all possible combinations of different assumptions on how the oracles deal with not being able to compute the function with arbitrary accuracy. We also discuss the introduction of constraints (or, more generally, of easy components) and use of (partly) aggregated models.

Key words. nonsmooth optimization, bundle methods, inexact function evaluation, incremental approach

AMS subject classifications. 90C26, 65K05

DOI. 10.1137/16M1089897

1. Introduction.

We are concerned with the following minimization problem:

$$\min \left\{ f(x) = \sum_{k\in\mathcal{K}} f^k(x) \ : \ x \in X \right\} \ , \tag{1.1}$$

where $\mathcal{K}$ is a finite index set, $X \subseteq \mathbb{R}^n$ is closed, convex, and "easy" in a sense specified later, and each *component* $f^k : \mathbb{R}^n \to \mathbb{R}$ of f is convex but possibly nondifferentiable. Customarily, we assume that each f^k is available through an (approximate) *oracle*, i.e., a procedure which, given x, returns (approximate) information about the value of $f(x)$ and the first-order behavior of f at x; our specific definition is given in (2.2). Our development hinges on a somewhat stronger assumption than usual, though:

$$\text{each } f^k \text{ is } \textit{globally Lipschitz} \text{ on } \mathbb{R}^n \text{ with } \textit{known} \text{ Lipschitz constant } L^k. \tag{1.2}$$

We are especially motivated by the case of a block-structured problem

$$\sup \left\{ \sum_{k\in\mathcal{K}} c^k u^k \ : \ \sum_{k\in\mathcal{K}} A^k u^k = b \ \ , \ \ u^k \in U^k \quad k \in \mathcal{K} \right\} \ , \tag{1.3}$$

where f is the *Lagrangian function* w.r.t. the "complicating" constraints that link together blocks of variables that would otherwise be independent, i.e., with

$$f^k(x) := \sup \left\{ \ (c^k - xA^k)u^k \ : \ u^k \in U^k \ \right\} \ , \tag{1.3$_x^k$}$$

$$f(x) = xb + \sum_{k\in\mathcal{K}} f^k(x) \ \ . \tag{1.4}$$

[*] Received by the editors August 17, 2016; accepted for publication (in revised form) October 11, 2017; published electronically February 6, 2018.
http://www.siam.org/journals/siopt/28-1/M108989.html

Funding: The work of the authors was supported by the COST Action TD1207, and financial support from the Gaspard-Monge program for Optimization and Operations Research (PGMO) project "Consistent Dual Signals and Optimal Primal Solutions."

[†] EDF R&D OSIRIS, F-91120 Palaiseau Cedex, France (wim.van-ackooij@edf.fr).

[‡] Dipartimento di Informatica, Università di Pisa, 56127 Pisa, Italy (frangio@di.unipi.it).

For each $k \in \mathcal{K}$, any optimal solution u^k_* of (1.3^k_x) provides the function value $f^k(x) = (c^k - xA^k)u^k_*$ and the subgradient $z^k = -A^k u^k_* \in \partial f^k(x)$. For such f, (1.2) is often true, for instance, because U^k is nonempty and compact and finite bounds $-\infty < \underline{u}^k \leq u^k \leq \bar{u}^k < \infty$ are known for each $u^k \in U^k$ (very often, $U^k \subseteq \{0,1\}^{n_k}$). Minimizing f solves the *Lagrangian dual* of (1.3), which has countless applications, e.g., [5, 6, 14, 17, 18, 20, 28, 31] among the many others. Typically (1.3) is "difficult," due to either being large-scale, $\mathcal{NP}$-hard, or both. Hence, computing f (i.e., computing a valid bound) is cheaper than solving (1.3), if only because it separates into $|\mathcal{K}|$ smaller subproblems. However it may still be costly, as each (1.3^k_x) may still be $\mathcal{NP}$-hard, $|\mathcal{K}|$ may be large, or both. Thus, finding ways to reduce the function evaluation cost may be useful. An attractive strategy is to compute f only *approximately* to within some *error* ε, an issue that has seen substantial interest of late [11, 12, 13, 15, 33, 36, 37]. In our sum-function context, this may actually mean two different things: that problems (1.3^k_x) are approximately solved, or that some of them are not solved at all. Approaches doing the latter are called *incremental*. However, in all proposals so far [10, 15, 21], avoiding the solution of some (1.3^k_x) is only possible at "bad" iterations where the f-value does not improve (a.k.a. *null steps* (NS)), while "good" iterations (*serious steps* (SS)) require that *all* the f^k are computed. This is basically due to the fact that, in order to prove that any $x \in X$ is approximately optimal to some accuracy ε, one has to compute its function value $f(x)$ with at least the same accuracy [9, Observation 2.7]. Yet, our development will clarify that what is really needed is an *upper bound* on $f(x)$.

Upper bounds on $f(x)$ are not directly mentioned in the literature about bundle methods, except in the recent [35] for a different context. They are indirectly used in the fundamental reference [12]; in particular, the *controllable lower oracle* there explicitly produces lower estimates only, but it has a *known* maximum error out of which worst-case upper estimates can be derived. This is used to define the *conservative decrease* [12, (5.8)], which is basically what we will use (cf. (3.4)), except that our upper estimates may be tighter than the worst-case ones. Hence, the controllable bundle method [12, Algorithm 5.4] and the asymptotically exact bundle method of [12, section 7.1.4] are very close to the methods we analyze here. However, in our analysis upper estimates take center stage: they are explicitly produced by the oracle, which can at some iteration produce *only* them. We show that availability of upper estimates allows one to refine somewhat the standard convergence analysis of inexact bundle methods; in particular, it allows one to produce *explicit a posteriori estimates of the quality of the obtained solution* and to define *three different forms of the crucial noise reduction (NR) step*, only one of which was previously known. As a consequence, we don't require the accuracy to be nonincreasing, as in [12, Remark 6.8], nor to be exactly zero when the function value is below a given target, as in [12, section 7.1.2], [21].

Also, explicitly considering upper estimates helps in providing the oracle(s) with a better description of the conditions that are needed from the returned information so that the optimization can proceed under the form of *two targets*, an upper and a lower one, and the accuracy. For the Lagrangian case (1.3)/(1.4), this might allow one to terminate early on the solution of problems (1.3^k_x). All these advantages actually apply to the general inexact case, even if the function is not a sum one, only provided the oracle produces explicit upper estimates.

Finally, and crucially for the inexact case, having upper estimates available also allows one to complement the usual *lower model*(s of the individual components f^k) of f, that traditionally drive the optimization process, with an *upper model* that provides *upper estimates* of $f(x)$ even if no oracle has ever been called at x. This has already been done in [1], but only on a small subset of the search space: exploiting (1.2) we extend the upper model to all of X. This is the fundamental technical idea that allows us to prove convergence without necessarily requiring that all components have been evaluated at SS.

This work is organized as follows. In section 2 we introduce the different ingredients of the algorithm, their rationale, and key notation. The algorithm is given in section 3, where we also discuss the convergence analysis in detail. In section 4 we extend the framework to three different classes of oracles that may not be able to provide information with arbitrary accuracy, and we conclude in section 5.

2. Ingredients of the algorithm.

We start by illustrating the main components that have to be assembled to form a complete algorithm.

2.1. The oracle.

For our development we extend the definition of the *inexact, informative, on-demand* oracle of [35]. Any oracle $\mathcal{O}^k$ for f^k, when called at some given $x \in X$, has to provide information about the function value and the first-order behavior of f^k at x. We require these to define a standard *lower linearization* of f^k, i.e., a *lower estimate* $\underline{f}^k \leq f^k(x)$ and a vector $z^k \in \mathbb{R}^n$ such that

$$f^k(\cdot) \geq \underline{f}^k + \left\langle z^k, \cdot - x \right\rangle . \tag{2.1}$$

In order to control the accuracy of $\underline{f}^k$ (and therefore of z^k), besides $x \in X$ our oracle inputs *three* parameters $-\infty \leq \underline{\mathtt{tar}}^k \leq \overline{\mathtt{tar}}^k \leq \infty$ (the *lower and upper targets*, with $\overline{\mathtt{tar}}^k > -\infty$ and $\underline{\mathtt{tar}}^k < \infty$) and $0 \leq \varepsilon^k \leq \infty$ (the *accuracy*), and provides

$$\left\{ \begin{array}{l} \left[\begin{array}{l} \text{(i) \textit{function value information}: \textit{two} values } \underline{f}^k \text{ and } \bar{f}^k \text{ s.t.} \\ \qquad -\infty \leq \underline{f}^k \leq f^k(x) \leq \bar{f}^k \leq \infty \quad \text{and} \quad \bar{f}^k - \underline{f}^k \leq \varepsilon^k ; \\ \text{(ii) \textit{first-order information}: if } \underline{f}^k > -\infty \text{, a } z^k \in \mathbb{R}^n \text{ s.t. (2.1) holds;} \end{array} \right. \\ \text{(iii) s.t. \textit{at least one} between } \bar{f}^k \leq \overline{\mathtt{tar}}^k \text{ and } \underline{f}^k \geq \underline{\mathtt{tar}}^k \text{ holds.} \end{array} \right. \tag{2.2}$$

We will denote by $\mathcal{O}^k(\underline{\mathtt{tar}}^k, \overline{\mathtt{tar}}^k, \varepsilon^k, x)$ a call to the oracle. It is easy to realize that it is always possible for the oracle to provide a correct answer, possibly at the cost of computing $f^k(x)$ with "infinite" accuracy. Also, requiring a finite accuracy $\varepsilon^k < \infty$ entails the restitution of *finite* bounds and a subgradient, even if $\overline{\mathtt{tar}}^k = \infty$ and $\underline{\mathtt{tar}}^k = -\infty$. Indeed, since (2.2(i)) then holds with $\varepsilon^k < \infty$, *both* $\bar{f}^k$ and $\underline{f}^k$ must be finite, so (2.2(ii)) together with (2.1) gives $z^k \in \partial_{\varepsilon^k} f^k(x)$. As a consequence the following holds.

LEMMA 2.1. *Under* (1.2), $\left\| z^k \right\| \leq L^k$ *for each* z^k *produced by* (2.2).

Proof. A z^k is produced whenever $\underline{f}^k > -\infty$, and hence $z^k \in \partial_\varepsilon f^k(x)$ for $\varepsilon = f^k(x) - \underline{f}^k < \infty$. It is immediate to prove by the definition that $\left\| z^k \right\| \leq L^k$ for any $z^k \in \partial_\varepsilon f^k(x)$ (or use [25, Proposition 4.1.2]). ☐

Our setting is therefore, at least initially, biased toward subproblems (1.3^k_x) that can be solved with arbitrary accuracy; however, the extension to more relaxed assumptions will be provided in section 4.

It may be worth remarking that three oracle parameters may seem somewhat redundant, especially as all other approaches in the literature only use one or two. For instance, if *both* $\bar{f}^k \leq \overline{\mathtt{tar}}^k$ *and* $\underline{f}^k \geq \underline{\mathtt{tar}}^k$, then $\bar{f}^k - \underline{f}^k \leq \overline{\mathtt{tar}}^k - \underline{\mathtt{tar}}^k$, and ε^k might be deemed useless. However, (2.2) is more flexible than previous definitions of approximate oracles in that it allows $\varepsilon^k = \infty$, whereby it then requires *only one* between $\bar{f}^k$ and $\underline{f}^k$ to be finite. This is particularly relevant if (1.3^k_x) is $\mathcal{NP}$-hard, since then computing $f^k(x)$ typically amounts to *three* different processes:

1. finding a "good enough" feasible solution $\bar{u}^k \in U^k$ by (arbitrarily complex) *heuristics*, providing both the lower bound $\underline{f}^k = c^k\bar{u}^k \leq \nu(1.3^k_x)$ and the first-order information $z^k = -A^k\bar{u}^k$, where $\nu(\cdot)$ denotes the optimal value of an optimization problem;
2. producing a "good enough" upper bound $\bar{f}^k \geq \nu(1.3^k_x)$ by the exact solution of some appropriate *relaxation* of the problem (or a feasible solution of an appropriate dual problem, if available), again an arbitrarily complex process, considering that the solution of (1.1) for (1.4) is often nothing but the computation of an upper bound on $\nu(1.3)$;
3. if $\underline{f}^k$ and $\bar{f}^k$ are not "close enough" (cf. (2.2)(i)), performing an arbitrary amount of branching and/or tightening (say, by valid inequalities) to make them so.

When, say, a general-purpose MILP solver is used to solve (1.3^k_x), these three processes are tightly integrated; however, they nonetheless remain conceptually distinct. Thus, any such solver typically produces candidate values $\underline{f}^k \leq \nu(1.3^k_x) \leq \bar{f}^k$ and gradually reduces the distance between the two. Given the three parameters in (2.2) it is easy to stop the solution process (e.g., via appropriate `callback` functions) as soon as the required conditions are satisfied. The parameters are not redundant to each other: while $\overline{\texttt{tar}}^k$ and $\underline{\texttt{tar}}^k$ set specific targets on $\nu(1.3^k_x)$, but are independent from one another, ε^k requires both bounds, but it makes no assumptions on where $\nu(1.3^k_x)$ lies. For $\varepsilon^k = \infty$, this means that the computation can be interrupted without having one of the two bounds at all. For instance, if $\bar{u}^k \in U^k$ is found such that $c^k\bar{u}^k \geq \underline{\texttt{tar}}^k$, one can entirely avoid computing any relaxation, since then $\bar{f}^k = \infty$ is allowed. Symmetrically, if a bound $\nu(1.3^k_x) \leq \bar{f}^k \leq \overline{\texttt{tar}}^k$ is obtained, one can stop without producing any $\bar{u}^k \in U^k$, and therefore any z^k, since $\underline{f}^k = -\infty$ is allowed. This is more flexible than all approaches proposed so far in the literature. For instance, [11, 35] for the nonsum case (the former a level method, the latter for a discrete case) produce upper estimates (the former implicitly, the latter explicitly) but only have one target and must always produce z^k. The recent [10] for the sum case does not set any target to the individual oracles, although a global (upper) target is set for f. Our setting therefore has the potential to reduce the number of iterations in which (1.3^k_x) is solved with high accuracy.

2.2. Lower models. In standard bundle methods, all components $k \in \mathcal{K}$ are evaluated at each point in a sequence $\{x_\ell\}$ of iterates, producing information z^k_ℓ and $\underline{f}^k_\ell$. This is then used to define aggregated information $z_\ell = \sum_{k\in\mathcal{K}} z^k_\ell$ and $\underline{f}_\ell = \sum_{k\in\mathcal{K}} \underline{f}^k_\ell$ satisfying (2.1) for the whole of f. For reasons to become clearer shortly (cf. (2.8)), one customarily replaces $\underline{f}_\ell$ with $\alpha_\ell = \langle z_\ell, x_\ell\rangle - \underline{f}_\ell$ to define the *(aggregated lower) bundle* $\mathcal{B} = \{\,(z_i, \alpha_i)\,\}$, which is the basis of the (*aggregated*) *cutting plane model*

$$\check{f}_{\mathcal{B}}(x) = \max\{\, \langle z_i, x\rangle - \alpha_i \;:\; i \in \mathcal{B}\,\} \tag{2.3}$$

(with the useful shorthand "$i \in \mathcal{B}$" for "$(z_i, \alpha_i) \in \mathcal{B}$"). Upon first reading one may assume $i = \ell$, which, via (2.1), immediately proves that $\check{f}_{\mathcal{B}} \leq f$, i.e., $\check{f}_{\mathcal{B}}$ is a *lower model* of f. However, in general the pairs in $\mathcal{B}$ are not directly correlated with the iterates, as we shall see, whence the different index. While $\check{f}_{\mathcal{B}}$ is not the only possible (lower) model of f [2, 3, 32], our development only uses cutting plane models.

The issue with $\check{f}_{\mathcal{B}}$ is that it requires to compute all components at all iterations, which is precisely what we want to avoid. For this it is necessary to consider *individual lower models* for each component, i.e.,

$$\check{f}^k_{\mathcal{B}}(x) = \max\left\{\ \langle z^k_i, x\rangle - \alpha^k_i \ :\ (z^k_i, \alpha^k_i) \in \mathcal{B}^k\ \right\} \leq f^k(x) \tag{2.4}$$

depending on *individual (lower) bundles* $\mathcal{B}^k = \{\,(z^k_i, \alpha^k_i = \langle z^k_i, x_i\rangle - \underline{f}^k_i)\,\}$. It will be appropriate to still refer to $\mathcal{B} = [\,\mathcal{B}^k\,]_{k\in\mathcal{K}}$ as "the lower bundle," with the distinction between this and the aggregated version easily made by the context. In our algorithm, not all individual lower bundles will be updated at all iterations ℓ. Indeed, not only we will strive not to call the oracles for all components at every iterate, but even if called, an oracle may return $\underline{f}^k = -\infty$, and therefore no z^k, and this may still be enough for the algorithm to proceed. This is not the only reason why i and ℓ are different indices: the well-known *aggregation technique* (cf. (3.9)) can be used to produce pairs (z^k_i, α^k_i) that bear no direct relationship with any specific x_ℓ.

In the analysis, whenever possible we will not distinguish between (2.3) and the *disaggregated lower model* $\check{f}_{\mathcal{B}}(x) = \sum_{k\in\mathcal{K}} \check{f}^k_{\mathcal{B}}(x) \leq f(x)$. Note that here we are disregarding possible "simple" terms in f, like the linear one in (1.4) associated to the right-hand-side of the relaxed coupling constraints, or constraints $x \in X$; these will be dealt with in section 3.5. It has to be remarked that the choice between aggregated and disaggregated lower models is by no means inconsequential. In fact, while using (2.4) is well-known to improve, often substantially, the convergence speed of the algorithm, it may also come at a significant cost in terms of solution time of the master problem (cf. section 2.4). So, while (2.4) is often preferable (e.g., [19]), there are cases where (2.3) is more efficient. In the recent [38], for instance, *both* models are actually used depending on the type of iteration (NS or SS). While we initially present our approach with the use of disaggregated models, different choices will be briefly discussed in section 3.4.

2.3. Upper models. The fact that our oracles (2.2) explicitly produce upper estimates $\bar{f}^k \geq f^k(x)$ also allows us to define *upper models* of f^k. For each $k \in K$ we define the (*individual*) *upper bundle* $\mathcal{P}^k = \{\,(x_i, \bar{f}^k_i)\,\}$ with $f^k(x_i) \leq \bar{f}^k_i < \infty$. Then, the upper bundle is $\mathcal{P} = [\,\mathcal{P}^k\,]_{k\in\mathcal{K}}$; although it may make conceptually sense to define an aggregated upper bundle made of pairs $(x_i, \bar{f}_i = \sum_{k\in\mathcal{K}} \bar{f}^k_i)$, we won't pursue this direction. As for $\mathcal{B}^k$, while some $(x_i, \bar{f}^k_i) \in \mathcal{P}^k$ will have $x_i = x_\ell$ and $\bar{f}^k_i$ as the oracle output when called on x_ℓ, in general not all upper bundles will be updated at each iterate, both because some $k \in \mathcal{K}$ may not be evaluated, and because, even if it is, the oracle may report $\bar{f}^k = \infty$. Furthermore, *aggregation techniques* akin to those known for the lower bundles (cf. (3.7)) can add pairs $(x_i, \bar{f}^k_i)$, where x_i is not any of the previous (or future) iterates x_ℓ. Yet, using the standard shorthand, it is clear that

$$\dot{f}^k_{\mathcal{P}}(x) = \inf\left\{\ \sum_{i\in\mathcal{P}^k} \bar{f}^k_i \theta^k_i \ :\ \sum_{i\in\mathcal{P}^k} x_i \theta^k_i = x\ ,\ \theta^k \in \Theta^k\ \right\} \geq f(x) \tag{2.5}$$

for all $x \in X$ with Θ^k the unitary simplex in appropriate dimension. However, (2.5) may admit no solution: $\dot{f}^k_{\mathcal{P}}(x) = \infty$ for $x \notin \bar{X}^k_{\mathcal{P}} = \operatorname{conv}(\{\,x_i\ :\ i \in \mathcal{P}^k\,\})$. Furthermore, since $\bar{f}^k_i \gg f^k(x_i)$ may happen, there is no guarantee that $\dot{f}^k_{\mathcal{P}}$ satisfies

(1.2) with L^k. Yet, exploiting (1.2) we can improve $\dot{f}^k_{\mathcal{P}}$ by defining

$$\begin{aligned} \hat{f}^k_{\mathcal{P}}(x) &= \inf\{\, \dot{f}^k_{\mathcal{P}}(w) + L^k\|x-w\|_2 \ : \ w \in X \,\} \\ &= \inf\left\{ \sum_{i\in\mathcal{P}^k} \bar{f}^k_i\theta^k_i + L^k\|s^k\|_2 \ : \ \sum_{i\in\mathcal{P}^k} x_i\theta^k_i + s^k = x \ , \ \theta^k \in \Theta^k \right\} \ . \end{aligned} \tag{2.6}$$

Clearly, $f^k(x) \leq \hat{f}^k_{\mathcal{P}}(x) < \infty$: (1.2) implies that $f^k(x) \leq f^k(w) + L^k\|x-w\| \leq \dot{f}^k_{\mathcal{P}}(w) + L^k\|x-w\|$ for any $x \in X$ and arbitrary w. This kind of regularization has been studied in [24, section XI.3.4]; Proposition XI.3.4.5 there proves that $\hat{f}^k_{\mathcal{P}}(x) = \dot{f}^k_{\mathcal{P}}(x) \iff \exists z^k \in \partial\dot{f}^k_{\mathcal{P}}(x)$ such that $\|z^k\| \leq L^k$. As previously remarked this may not happen, not even for $x \in \mathit{int}\,\bar{X}^k_{\mathcal{P}}$; thus, $\hat{f}^k_{\mathcal{P}}$ both extends and (potentially) improves $\dot{f}^k_{\mathcal{P}}$. As (2.6) shows, $\hat{f}^k_{\mathcal{P}}$ can be computed "cheaply"; thus, the same holds for the global upper model $\hat{f}_{\mathcal{P}}(x) = \sum_{k\in\mathcal{K}} \hat{f}^k_{\mathcal{P}}(x) \geq f(x)$. Clearly, $\hat{f}_{\mathcal{P}}(x) < \infty$ for all $x \in X$, whenever $\mathcal{P}^k \neq \emptyset$ for all $k \in \mathcal{K}$.

Introducing $\hat{f}_{\mathcal{P}}$ is instrumental in avoiding the asymmetry that was present in all previous analyses of incremental bundle approaches. Producing *lower estimates* on $f(x)$ has always been easy, since $\check{f}_{\mathcal{B}}(x) < \infty$ for all $x \in X$. However, proving for a given x that $z \in \partial_\varepsilon f(x)$ requires actually producing an *upper bound* $\bar{f} \geq f(x)$, i.e., some $\bar{f}^k \geq f^k(x)$ for *all* $k \in \mathcal{K}$. This is why all incremental bundle approaches so far have required computing *all* the components, at least in selected iterations. As we will see, the availability of $\hat{f}^k_{\mathcal{P}}(x) < \infty$ will allow us to relax this strong requirement.

2.4. The master problem. The main use of the lower model is to drive the search for the next iterate. In the standard cutting-plane (CP) approach this would be the minimum of $\check{f}_{\mathcal{B}}$ over X, but this is well-known to suffer from *instability* issues. Bundle methods try to avoid that by ensuring that the next iterate lies in an appropriate (most often, implicitly defined) neighborhood of a suitably chosen point $\bar{x} \in X$, called the *stability center.* In particular, *proximal* bundle methods such as those studied here obtain this by solving the *stabilized master problem*

$$x_+ = \operatorname{argmin}\{\, \check{f}_{\mathcal{B}}(x) + \tfrac{1}{2t}\|x-\bar{x}\|^2 \,\} \ . \tag{2.7}$$

The *stabilizing term* $\|x-\bar{x}\|^2/(2t)$, governed by the *stabilization parameter* $t > 0$, ensures that x_+ will be "near" $\bar{x}$, thereby limiting the violent oscillations of the iterates typical of the unstabilized CP method, and frequently held responsible for its slow convergence. Additionally, it ensures that (2.7) is always bounded from below.

The introduction of the stability center $\bar{x}$ suggests to translate the lower models (2.4) using $\bar{x}$ as the origin. In the context of *exact* bundle methods, it is customary to also set the origin of the objective axis to $f(\bar{x})$, i.e., the reference value against which the decrease of f (and of $f_{\mathcal{B}}$) is measured. In *inexact* bundle methods it is necessary to choose a reference value to replace $f(\bar{x})$; in [12] this is the (lower) *level* $l \leq f(\bar{x})$, which is used to compute the crucial *model decrease* [12, equation (3.8)]. That general choice allows us to analyze convergence of many different inexact bundle variants, among which those with upper oracles that do not even guarantee (2.1). However, the model decrease using l as the reference can then be compared with an *effective decrease* that uses instead an upper estimate $\bar{f} \geq f(\bar{x}) \geq l$ as reference value [12, equation (5.8)]. We take a different path by defining the *linearization error* of z^k_i w.r.t. $\bar{x}$ and $\mathcal{P}$ as

$$\alpha^k_i(\bar{x},\mathcal{P}) = \hat{f}^k_{\mathcal{P}}(\bar{x}) - [\, \underline{f}^k_i + z^k_i(\bar{x}-x_i) \,] \ , \tag{2.8}$$

i.e., using the upper estimate $\hat{f}_{\mathcal{P}}(\bar{x}) \geq f(\bar{x})$ as the reference for measuring both decreases. Although the difference with previous approaches may look minor, in fact this decision has a significant impact on the convergence arguments. Clearly, (2.8) unavoidably hinges on availability of upper estimates. In our incremental setting this requires the combination of two assumptions:

1. the informative oracle (2.2) producing them;
2. the definition of the upper model (2.6) allowing to transport them to points where the oracle has not been called, which in turn requires (1.2).

However, in the nonincremental case—that is, either one single oracle, or calling all oracles at all iterates—the upper model is not necessary, and so neither is (1.2); in other words, the upper bundle $\mathcal{P}$ returns to be just the (upper estimate on the) function value(s) at $\bar{x}$, as in standard approaches. As we shall see, a large part of our analysis remains valid, and interesting, for the general case of an inexact bundle method with the informative oracle (2.2), even if no incremental strategy is employed. Indeed, it is only the availability of upper estimates for all $k \in \mathcal{K}$, however they be obtained, that allows us to use (2.8); its immediate consequence, using convexity of f^k, $\underline{f}_i^k \leq f^k(x_i)$ and $\hat{f}^k_{\mathcal{P}}(\bar{x}) \geq f^k(\bar{x})$, is

$$(2.9) \qquad z_i^k \text{ satisfies (2.1)} \quad \Longrightarrow \quad \alpha_i^k(\bar{x}, \mathcal{P}) \geq 0 \quad \Longrightarrow \quad z_i^k \in \partial_{\alpha_i^k(\bar{x},\mathcal{P})} f^k(\bar{x}) \enspace .$$

That is, $\alpha_i^k(\bar{x}, \mathcal{P})$ measures the accuracy of z_i^k as first-order information of f^k in $\bar{x}$. This measure is easily recomputed if the stability center is updated to any other $\tilde{x}$ (with corresponding $\tilde{f}^k \geq f^k(\tilde{x})$) by the well-known *information transport property*

$$(2.10) \qquad \alpha_i^k(\tilde{x}, \mathcal{P}) = z_i^k(\, \bar{x} - \tilde{x}\,) + \alpha_i^k(\bar{x}, \mathcal{P}) + (\, \hat{f}^k_{\mathcal{P}}(\tilde{x}) - \hat{f}^k_{\mathcal{P}}(\bar{x})\,) \enspace (\geq 0\,) \enspace .$$

Since $\bar{x}$ and $\mathcal{P}$ are usually clear from the context, to alleviate the notation it is customary to use just α_i^k in place of $\alpha_i^k(\bar{x}, \mathcal{P})$. Care will have to be taken, since in the standard analysis the linearization errors only change when $\bar{x}$ does (i.e., during SS), whereas in our case they can change even if $\bar{x}$ doesn't. This dependency will be appropriately discussed when needed. It is also important to remark that, with our definition, the linearization errors "take into account the *gap* $\hat{f}^k_{\mathcal{P}}(\bar{x}) - \check{f}^k_{\mathcal{B}}(\bar{x}) \geq 0$" between the upper and lower models at $\bar{x}$. Indeed, plug $\check{f}^k_{\mathcal{B}}(\bar{x}) \geq \underline{f}_i^k + z_i^k(\bar{x} - x_i)$ into (2.8) to obtain

$$(2.11) \qquad \hat{f}^k_{\mathcal{P}}(\bar{x}) - \check{f}^k_{\mathcal{B}}(\bar{x}) = \min\{\, \alpha_j^k \;:\; j \in \mathcal{B}^k \,\} \leq \alpha_i^k \quad \forall i \in \mathcal{B}^k \enspace .$$

For fixed $\bar{x}$ and $\mathcal{P}$, the (*disaggregated*, cf. (2.4)) master problem (2.7) can then be written

$$(2.12) \qquad \min \left\{ \sum_{k \in \mathcal{K}} v^k + \frac{1}{2t} \|d\|^2 \;:\; v^k \geq z_i^k d - \alpha_i^k \quad i \in \mathcal{B}^k \enspace , \enspace k \in \mathcal{K} \right\}$$

and its optimal solution d_* gives $x_+ = \bar{x} + d_*$, where the oracle is typically called. Note that $v^k \geq \check{f}^k_{\mathcal{B}}(x_+) - \hat{f}^k_{\mathcal{P}}(\bar{x})$ (cf. (2.8)), and therefore for the optimal value v_*^k

$$(2.13) \qquad v_* = \check{f}_{\mathcal{B}}(x_+) - \hat{f}_{\mathcal{P}}(\bar{x}) = \sum_{k \in \mathcal{K}} \left(v_*^k = \check{f}^k_{\mathcal{B}}(x_+) - \hat{f}^k_{\mathcal{P}}(\bar{x}) \right) \enspace ,$$

a relationship that will be crucial later on. The usefulness of defining the linearization errors precisely via (2.8) lies in the fact that the *dual* of (2.12) is

$$(2.14) \qquad \min \left\{ \frac{1}{2} t \left\| \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{B}^k} z_i^k \theta_i^k \right\|^2 + \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{B}^k} \alpha_i^k \theta_i^k \;:\; \theta^k \in \Theta^k \enspace , \enspace k \in \mathcal{K} \right\} \enspace ,$$

where Θ^k is the unitary simplex of dimension $|\mathcal{B}^k|$. Standard duality theory guarantees that $\nu(2.12) = -\nu(2.14)$, and the relationships

$$d_* = -tz_* \quad , \quad v_* = -t\,\|z_*\|^2 - \alpha_* = \sum_{k\in\mathcal{K}} (d_* z^k_* - \alpha^k_*) = \sum_{k\in\mathcal{K}} v^k_* \tag{2.15}$$

between the primal and dual optimal solutions, where

$$z^k_* = \sum_{i\in\mathcal{B}^k} z^k_i \theta^k_{*,i} \quad , \quad \alpha^k_* = \sum_{i\in\mathcal{B}^k} \alpha^k_i \theta^k_{*,i} \quad , \quad z_* = \sum_{k\in\mathcal{K}} z^k_* \quad , \quad \alpha_* = \sum_{k\in\mathcal{K}} \alpha^k_* \tag{2.16}$$

"translate in the (z,α)-space" the dual optimal solution θ_* of (2.14). Consequently, we define

$$\Delta_* := \nu(2.14) = (t/2)\,\|z_*\|^2 + \alpha_* = -\nu(2.12) = -v_* - \|d_*\|^2/(2t)\,(>0) \ . \tag{2.17}$$

These relationships are crucial in the analysis of the method, since their obvious consequence

$$z^k_* \in \partial_{\alpha^k_*} f^k(\bar{x}) \quad k \in \mathcal{K} \qquad \Longrightarrow \qquad z_* \in \partial_{\alpha_*} f(\bar{x}) \tag{2.18}$$

(cf. (2.16)) formally proves that $z_* = 0$ ($\Longrightarrow d_* = 0$) and $\alpha_* = 0$ imply that $0 \in \partial f(\bar{x})$, i.e., $\bar{x}$ is optimal. In practice one therefore stops when $\|z_*\|$ and α_* are "small." Hence, whenever one does not stop, $v_* < 0$ (cf. (2.15)); in particular it is *not* small, i.e., d_* is a (significant) *descent direction.* Note that this is *not* true for the individual components, i.e., $v^k_* > 0$ may happen for some (but not all) k. The *predicted descent* (2.13) is crucial in the analysis: its componentwise characterization

$$\check{f}^k_{\mathcal{B}}(x_+) = \hat{f}^k_{\mathcal{P}}(\bar{x}) + d_* z^k_* - \alpha^k_* = \hat{f}^k_{\mathcal{P}}(\bar{x}) + v^k_* \tag{2.19}$$

(cf. (2.13) and (2.15)) shows that v_* also "includes the gap $\hat{f}_{\mathcal{P}}(\bar{x}) - \check{f}_{\mathcal{B}}(\bar{x}) \geq 0$," since (2.11) implies $\alpha^k_* \geq \hat{f}^k_{\mathcal{P}}(\bar{x}) - \check{f}^k_{\mathcal{B}}(\bar{x})$, which in turn (using (2.15) for the first inequality) gives

$$-v_* \geq \alpha_* = \sum_{k\in\mathcal{K}} \alpha^k_* \geq \sum_{k\in\mathcal{K}} \big(\, \hat{f}^k_{\mathcal{P}}(\bar{x}) - \check{f}^k_{\mathcal{B}}(\bar{x}) \,\big) = \hat{f}_{\mathcal{P}}(\bar{x}) - \check{f}_{\mathcal{B}}(\bar{x}) \ . \tag{2.20}$$

3. The base algorithm.

We now describe a first version of the algorithm that mimics as closely as possible those for the exact case, and discuss its convergence. To alleviate the heavy notation we will use shorthand whenever possible, for instance, $\hat{f}_\ell(\cdot)$ for $\hat{f}_{\mathcal{P}_\ell}(\cdot)$ and $\check{f}_\ell(\cdot)$ for $\check{f}_{\mathcal{B}_\ell}(\cdot)$, ℓ being the iteration index. The reference point where f is estimated is $\bar{x}_\ell$, so it makes sense to use $\check{f}_\ell = \check{f}_\ell(\bar{x}_\ell)$ and $\hat{f}_\ell = \hat{f}_\ell(\bar{x}_\ell)$. We will also dismiss ℓ altogether when no confusion arises, as in $\bar{x}$ and x_+ for $\bar{x}_\ell$ and $x_{\ell+1}$, $\hat{f}^k$, $\check{f}^k$, $\hat{f}^k_+$, and $\check{f}^k_+$ for $\hat{f}^k_\ell$, $\check{f}^k_\ell$, $\check{f}^k_\ell(x_+)$ and $\hat{f}^k_\ell(x_+)$, respectively. Each time we obtain function value estimates from the oracle we ensure they are consistent with $\hat{f}^k$ and $\check{f}^k$ by

$$\bar{f}^k_+ := \min\big\{\, \bar{f}^k_+ \,,\, \hat{f}^k_{\mathcal{P}}(x_+) \,\big\} \quad , \quad \underline{f}^k_+ := \max\big\{\, \underline{f}^k_+ \,,\, \check{f}^k_{\mathcal{B}}(x_+) \,\big\} \ . \tag{3.1}$$

Minimal care in the handling of $\mathcal{B}^k$ and $\mathcal{P}^k$ will therefore ensure that

$$-\infty < \check{f}^k_+ = \bar{f}^k + v^k_* \leq \underline{f}^k_+ \leq f^k_+ \leq \bar{f}^k_+ \leq \hat{f}^k_+ < \infty \ , \tag{3.2}$$

so that (3.1) always allows us to assume $-\infty < \underline{f}^k_+ \leq \bar{f}^k_+ < \infty$, even though $\mathcal{O}^k$ has not been called yet (or has produced infinite values). It is also necessary to estimate the linearization error of z^k w.r.t. $\bar{x}$, whenever it was produced: the fact that $x_+ = \bar{x} + d_*$

gives, using (2.8)

$$\alpha^k = \hat{f}^k - \underline{f}_+^k + z^k d_* \ . \tag{3.3}$$

We will frequently need to restrict sums to subsets $\mathcal{H} \subseteq \mathcal{K}$ of the components, e.g., as in

$$\bar{f}_+^{\mathcal{H}} = \sum_{k \in \mathcal{H}} \bar{f}_+^k \quad , \quad \underline{f}_+^{\mathcal{H}} = \sum_{k \in \mathcal{H}} \underline{f}_+^k \quad , \quad z^{\mathcal{H}} = \sum_{k \in \mathcal{H}} z^k \quad , \quad \alpha^{\mathcal{H}} = \sum_{k \in \mathcal{H}} \alpha^k \ .$$

We will use "$-\mathcal{H}$" to refer to (sums over) the complement of $\mathcal{H}$, i.e., $\mathcal{K} \setminus \mathcal{H}$; taking "$k$" to mean "$\{ k \}$", "$-k$" then has to be read as $\mathcal{K} \setminus \{ k \}$.

3.1. Presentation of the algorithm. We will present the pseudocode of our algorithm in two parts. The "main loop" described next is as close as possible to that of standard bundle methods, so that we can prove its convergence with basically the very same arguments. This requires the full power of oracle (2.2), i.e., the ability to solve (1.3_x^k) with arbitrary accuracy; relaxing this assumption is analyzed in section 4. The main loop is parametric on the exact way in which the oracles are called, possibly repeatedly, on a subset of components in order to attain the crucial relationships required to ensure convergence (cf. (3.5)/(3.6)); this is the task of the "inner loop," presented and analyzed in section 3.3. We will refer to iterations in the main loop (indexed by ℓ) as *outer iterations*, to distinguish them from the *inner iterations* performed within the inner loop. The pseudocode of the main loop is

```
0 (Input and initializations) input the algorithmic parameters $\delta_1 \geq 0$, $\delta_2 \geq 0$, and $0 < m_1 < m_2 < 1$; set the iteration counter $\ell \hookleftarrow 1$; choose arbitrarily $\bar{x}_1$ and $t_1 > 0$; for all $k \in \mathcal{K}$, call $\mathcal{O}^k(-\infty, \infty, \varepsilon^k, \bar{x}_1)$ with arbitrary $0 \leq \varepsilon^k < \infty$, collect $-\infty < \underline{f}^k \leq (f^k(\bar{x}_1) \leq) \bar{f}^k < \infty$ and $z^k$; set $\mathcal{P}_1^k \hookleftarrow \{ (\bar{x}_1, \bar{f}^k) \}$, $\mathcal{B}_1^k \hookleftarrow \{ (z^k, \alpha^k(\bar{x}_1, \mathcal{P})) \}$;
1 (Master problem) solve (2.12)/(2.14) for the optimal solutions $d_{*,\ell}$, $v_{*,\ell}^k$, $\theta_{*,\ell}^k$, $z_{*,\ell}^k$, and $\alpha_{*,\ell}^k$;
2 (Stopping condition) if $\|z_{*,\ell}\| \leq \delta_1$, and $\alpha_{*,\ell} \leq \delta_2$, then stop else compute $\Delta_{*,\ell} \hookleftarrow t_\ell \|z_{*,\ell}\|^2 / 2 + \alpha_{*,\ell}$; $x_{\ell+1} \hookleftarrow \bar{x}_\ell + d_{*,\ell}$; $\check{f}_\ell(x_{\ell+1}) = \hat{f}_\ell(\bar{x}_\ell) + v_{*,\ell}$; define the global upper and lower targets
    (3.4)  $\overline{\mathtt{tar}}_\ell \hookleftarrow \check{f}_\ell(x_{\ell+1}) - m_2 v_{*,\ell}$ ,  $\underline{\mathtt{tar}}_\ell \hookleftarrow \check{f}_\ell(x_{\ell+1}) + m_1 \Delta_{*,\ell}$
  and set the global accuracy $\varepsilon_\ell \hookleftarrow \overline{\mathtt{tar}}_\ell - \underline{\mathtt{tar}}_\ell = -m_2 v_{*,\ell} - m_1 \Delta_{*,\ell}$;
3 (Inner loop) call the inner loop with parameters $\mathcal{B}_\ell$, $\mathcal{P}_\ell$, $\bar{x}_\ell$, $x_{\ell+1}$, $\varepsilon_\ell$, $v_{*,\ell}^k$ for each $k \in \mathcal{K}$, $\Delta_{*,\ell}$, $m_1$, and $m_2$; let $\mathcal{B}_{\ell+1}$, $\mathcal{P}_{\ell+1}$ be those output by the inner loop;
4 (NS/SS test) check the conditions
    (3.5)  $\hat{f}_{\ell+1}(x_{\ell+1}) \leq \overline{\mathtt{tar}}_\ell$,
    (3.6)  $\check{f}_{\ell+1}(x_{\ell+1}) \geq \underline{\mathtt{tar}}_\ell$;
  if (3.5) holds and (3.6) does not, then perform SS, i.e., $\bar{x}_{\ell+1} \hookleftarrow x_{\ell+1}$; if (3.6) holds and (3.5) does not, then perform NS: $\bar{x}_{\ell+1} \hookleftarrow \bar{x}_\ell$; if both (3.5) and (3.6) hold, then choose arbitrarily to perform either an SS or an NS;
5 (Bookkeeping) appropriately update $\mathcal{P}_{\ell+1}$ and $\mathcal{B}_{\ell+1}$ and select $t_{\ell+1}$; $\ell \hookleftarrow \ell + 1$; go to Step 1.
```

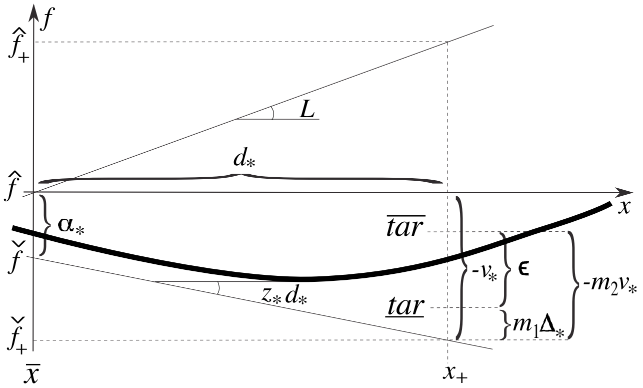

FIG. 1. *Illustration of the main quantities in the SS/NS decision.*

A few initial remarks are in order. The fact that $0 \leq \Delta_{*,\ell} \leq -v_{*,\ell}$ (cf. (2.17)) and $0 < m_1 < m_2 < 1$ imply $\check{f}_\ell(x_{\ell+1}) < \underline{\mathtt{tar}}_\ell < \overline{\mathtt{tar}}_\ell < \hat{f}_\ell(\bar{x}_\ell)$, and therefore $0 < \varepsilon_\ell < \infty$. Figure 1 pictorially illustrates the relationships between the main quantities involved in the crucial SS/NS decision, which will be useful while discussing convergence of the main loop in section 3.2. While we will strive to closely follow standard convergence arguments for (proximal) bundle methods, there will be some significant differences with the exact case. A striking one is that, in Step 4, it may well happen that *both* (3.5) and (3.6) hold true, which is quite unlike previous proposals; this might actually happen when evaluating the case depicted in Figure 1. Hence, the algorithm may have the choice between performing an SS and an NS, which may be useful in practice. Note that, at least until section 4, we will assume that *at least* one of the two conditions holds; cf. Assumption 1.

Another phenomenon that is specific of our setting is the fact that computing some $\bar{f}^k$ at $x_{\ell+1}$ and adding the corresponding pair $(x_{\ell+1}, \bar{f}^k)$ to $\mathcal{P}^k_\ell$ may, through (2.6), *decrease* the upper estimate at the stability center $\bar{x}_\ell$. In other words, one might have $\Delta\hat{f}^k_\ell = \hat{f}^k_\ell(\bar{x}_\ell) - \hat{f}^k_{\ell+1}(\bar{x}_\ell) > 0$, in which case all the α^k_i would decrease (by $\Delta\hat{f}^k_\ell$) within the inner loop. Yet, the algorithm computes the crucial targets (3.4) (and, similarly, (3.15)) before any oracle call and it does not revise them within the inner loop, so that the crucial targets in (3.5) and (3.6) only depend on the value of the upper model at the beginning of each outer iteration. Conversely, at least one among $\hat{f}_{\ell+1}(x_{\ell+1}) < \hat{f}_\ell(x_{\ell+1})$ and $\check{f}_{\ell+1}(x_{\ell+1}) > \check{f}_\ell(x_{\ell+1})$ holds because $\mathcal{B}_{\ell+1}$ and $\mathcal{P}_{\ell+1}$ include the new information generated during the inner loop.

Avoiding updating the main algorithmic thresholds during inner iterations has somewhat counterintuitive consequences that are worth commenting upon: the algorithm may perform "fake" SS or NS. Indeed, assume that the algorithm performs an SS—i.e., (3.5) holds—in an iteration where $\Delta\hat{f}^k_\ell > 0$, i.e., $\hat{f}^k_{\ell+1} < \hat{f}^k_\ell$. Clearly, the "true" decrease of f due to moving the stability center to $x_{\ell+1}$ is less than what was originally estimated using the initial value of $\hat{f}^k_\ell$, and it therefore may be less than what (3.5) requires. Yet, this is not really an issue, as $\hat{f}^k_{\ell+1}$ is indeed "significantly smaller" than $\hat{f}^k_\ell$: that part of the decrease could have also been obtained by keeping $\bar{x}_{\ell+1} = \bar{x}_\ell$ is, in the end, irrelevant. An extreme form of this phenomenon can be seen in the case that $d_{*,\ell} = 0 \Longrightarrow x_{\ell+1} = \bar{x}_\ell$, and still the algorithm does not stop. This occurrence, which cannot happen in exact bundle methods, implies that $\alpha_{*,\ell} \gg 0$: that is, $\bar{x}_\ell$ is the best estimate of the minimum of f provided by the lower model $\check{f}_\ell$, but there is a large gap between $\hat{f}_\ell$ and $\check{f}_\ell$ (which is "comprised" in the α_i^k and therefore in $\alpha_{*,\ell}$; cf. (2.11)/(2.20)). As in [9], this just results in the oracle being called again in $x_{\ell+1} = \bar{x}_\ell$, resulting in improved function estimates and, very possibly, ultimately yielding a fake SS that is actually only improving the upper estimate at $\bar{x}_\ell$. This may even happen infinitely many times (say, if $\bar{x}_1$ is optimal and $z^k = 0$ throughout): basically, instead of asking to the $\mathcal{O}^k$ exact information from the start, one is continuously asking them more and more refined ones. Analogously, the NS condition (3.6) is designed to ensure that the information introduced into $\mathcal{B}$ in the inner loop "decreases enough the value of $\nu(2.14)$" (cf. Lemma 3.11). Since $\alpha_{*,\ell}$ decreases if $\Delta\hat{f}_\ell^k > 0$, $\nu(2.14)$ would already decrease even if $\mathcal{B}$ would not changed: however, once again the desired reduction of $\nu(2.14)$ is achieved, and, as we shall see, this is enough to ensure convergence.

3.2. Convergence of the main loop. Global convergence of the algorithm is simple to prove using only slightly modified versions of the standard analysis of proximal bundle methods. Of course, it hinges on the fact that the inner loop ultimately works as intended, summarized by the following.

Assumption 1 (correctness of the inner loop). *At each iteration, the inner loop finitely terminates producing $\mathcal{P}_{\ell+1}^k$ and $\mathcal{B}_{\ell+1}^k$ such that at least one of* (3.5) *and* (3.6) *holds.*

In section 3.3 we will show that the inner loop can satisfy Assumption 1. This is quite obvious in the nonincremental case (say, only one oracle), and therefore our analysis applies to the general inexact case, although for that it gives only slightly refined results w.r.t. those already available in the literature. Indeed, we proceed along well-established guidelines; with $\delta_1 = \delta_2 = 0$, we prove

1. in an infinite sequence of SS, $\|z_{*,\ell}\| \to 0$ and $\alpha_{*,\ell} \to 0$, which means that the corresponding sequence $\{ \bar{x}_\ell \}$ is a minimizing one, if f is bounded from below;
2. in an infinite sequence of consecutive NS, $\|z_{*,\ell}\| \to 0$ and $\alpha_{*,\ell} \to 0$, which means that the (fixed) corresponding stability center $\bar{x}$ is optimal.

We start from the first point: let $\mathcal{L}_{\mathsf{SS}}$ be the index set of SS, and assume $|\mathcal{L}_{\mathsf{SS}}| = \infty$. Rules for "appropriate" handling of $\mathcal{P}^k$ and t at Step 5 are required. The management of $\mathcal{P}^k$ is obviously specific to our analysis, but for this we can exploit the analogue of the *aggregation technique* that is well-know for the (lower) bundle management (cf. (3.9)). The idea is that the optimal primal solution (θ_*^k, s_*^k) to (2.6) for $\bar{x}_\ell$ immediately provides the *aggregated primal pair* and the corresponding *poorman's upper bundles*

$$\mathcal{P}_*^k = \left\{ \, (\, x_*^k \, , \, \bar{f}_*^k \,) = \left(\, \sum_{i \in \mathcal{P}^k} x_i \theta_{*,i}^k + s_*^k \, , \, \sum_{i \in \mathcal{P}^k} \bar{f}_i^k \theta_{*,i}^k + L^k \|s_*^k\|_2 \right) \right\} \tag{3.7}$$

so that $\hat{f}_{\mathcal{P}_*}^k(\bar{x}_\ell) = \hat{f}_{\mathcal{P}}^k(\bar{x}_\ell)$. In plain words, one can substitute $\mathcal{P}$ with $\mathcal{P}_*$ while ensuring that the value of the upper model at the stability center does not change (increase). This immediately suggests the following.

Assumption 2 (upper model management).

(i) *For all ℓ and $k \in \mathcal{K}$, let $(\theta_{*,\ell}^k, s_{*,\ell}^k)$ be the optimal solution to* (2.6) *with $x = \bar{x}_\ell$ having produced the value $\hat{f}_\ell^k$. Then, either all the $(x_i, \bar{f}_i^k) \in \mathcal{P}_{\ell-1}^k$ such that $\theta_{*,i,\ell}^k > 0$ also belong to $\mathcal{P}_\ell^k$, or $(\, x_*^k \, , \, \bar{f}_*^k \,)$ of* (3.7) *belongs to $\mathcal{P}_\ell^k$.*

(ii) *If $|\mathcal{L}_{\mathsf{SS}}| = \infty$, then $\sum_{\ell \in \mathcal{L}_{\mathsf{SS}}} t_\ell = \infty$.*

The immediate consequence is that $\hat{f}_\ell \geq \hat{f}_{\ell+1}$ holds for all ℓ: (3.5) works for an SS, and Assumption 2(i) readily ensures that the optimal solution of (2.6) at iteration ℓ always remains feasible (albeit, if (3.7) has been used, in a "surrogate form") at iteration $\ell + 1$, so that the optimal value cannot increase during an NS. Note that, as discussed in section 3.1, $\hat{f}_\ell$ can actually *decrease* when an NS is performed; however, this just means that it is a fortiori nonincreasing. Clearly, standard (exact) bundle methods invariably resort to (3.7), only keeping the (exact) upper approximation value in the stability center, for *all* components. Instead, our general form of Assumption 2(i) allows us to have "richer" upper bundles, possibly leading to avoiding to compute some components at some $x_{\ell+1}$ provided that tight enough upper estimates are generated by "nearby" iterates. As far as management of t_ℓ is concerned, Assumption 2(ii) just requires that it does not converge to 0 "too fast," written in an abstract form (since it is impossible to know beforehand if $|\mathcal{L}_{\mathrm{SS}}| = \infty$); the easiest way to ensure that it holds is to impose $0 < \underline{t} \leq t_\ell$ for all ℓ. Global convergence of infinite sequences of SS is now at hand using standard arguments.

THEOREM 3.1. *Under Assumptions* 1 *and* 2, *if* $|\mathcal{L}_{\mathrm{SS}}| = \infty$, *then* $\lim_{\mathcal{L}_{\mathrm{SS}} \ni \ell \to \infty} \hat{f}_\ell = \liminf_{\mathcal{L}_{\mathrm{SS}} \ni \ell \to \infty} f(\bar{x}_\ell) = \nu(1.1)$. *If moreover,* (1.1) *admits any optimal solution, i.e.,* $\nu(1.1) > -\infty$, *then* $\|z_{*,\ell}\| \to 0$ *and* $\alpha_{*,\ell} \to 0$. *In addition, if* $\{t_\ell\}$ *is bounded from above, then the sequence* $\{\, \bar{x}_\ell \,\}$ *converges to the optimal solution of problem* (1.1).

Proof. Assumption 2(i) guarantees that the sequence $\{\, \hat{f}_\ell \,\}$ is nonincreasing, and therefore it has a limit $\hat{f}_\infty$. If $\hat{f}_\infty = -\infty$, then $\nu(1.1) \leq \lim_{\mathcal{L}_{\mathrm{SS}} \ni \ell \to \infty} f(\bar{x}_\ell) \leq \hat{f}_\infty = -\infty$, i.e., $\{\, \bar{x}_\ell \,\}_{\ell \in \mathcal{L}_{\mathrm{SS}}}$ is a minimizing sequence (which cannot converge to an optimal solution to (1.1) since there is none). For the case where $\hat{f}_\infty > -\infty$, since (3.5) holds at SS

$$
\begin{aligned}
& \hat{f}_{\ell+1} \leq \hat{f}_\ell + (1 - m_2) v_{*,\ell} \leq \hat{f}_\ell - (1 - m_2)\big(\, \Delta_{*,\ell} = t_\ell \|z_{*,\ell}\|^2 / 2 + \alpha_{*,\ell} \,\big) \\
(3.8) \quad & \Longrightarrow \sum_{\ell \in \mathcal{L}_{\mathrm{SS}}} \big(\, t_\ell \|z_{*,\ell}\|^2 / 2 + \alpha_{*,\ell} \,\big) < \infty \quad \Longrightarrow \quad t_\ell \|z_{*,\ell}\|^2 \to 0 \quad \text{and} \quad \alpha_{*,\ell} \to 0 \;\; .
\end{aligned}
$$

Using (3.8) together with (2.20) and the fact that $\hat{f}_\ell - f(\bar{x}_\ell) \leq \hat{f}_\ell - \check{f}_\ell$ gives $\lim_{\mathcal{L}_{\mathrm{SS}} \ni \ell \to \infty} \hat{f}_\ell - f(\bar{x}_\ell) = 0$: asymptotically, the upper estimate becomes tight. Hence, $\lim_{\mathcal{L}_{\mathrm{SS}} \ni \ell \to \infty} \hat{f}_\ell = \liminf_{\mathcal{L}_{\mathrm{SS}} \ni \ell \to \infty} f(\bar{x}_\ell)$. The rest follows from [7]: for $\ell \in \mathcal{L}_{\mathrm{SS}}$, one is performing a step of t_ℓ along $z_{*,\ell} \in \partial_{\alpha_{*,\ell}} f(\bar{x}_\ell)$ (cf. (2.18)), which is the basic scheme [7, equations (1.1)–(1.2)]. Since (3.8) are [7, equations (1.5)–(1.6)] and Assumption 2(ii) is [7, equation (1.4)], $\lim_{\mathcal{L}_{\mathrm{SS}} \ni \ell \to \infty} \hat{f}_\ell = \nu(1.1)$ comes from [7, Proposition 1.2]. The last part of the theorem follows as in [7, Theorem 4.4]: (3.8) and the assumption that $\{t_\ell\}$ is bounded give $\sum_{\ell \in \mathcal{L}_{\mathrm{SS}}} t_\ell (\, t_\ell \|z_{*,\ell}\|^2 / 2 + \alpha_{*,\ell} \,) < \infty$, i.e., [7, equation (2.8)], which is all that is needed to invoke [7, Proposition 1.3]. □

We remark that imposing boundedness of $\{t_\ell\}$ would be easy by requiring $t_\ell \leq \bar{t} < \infty$ in Assumption 2(ii): this would be free in this section but would create issues later on in section 4, and hence we avoid it. Also, convergence of iterates requires no hypotheses on t_ℓ if, e.g., the level sets of f are compact. None of this is particularly relevant, so we now turn to the case $|\mathcal{L}_{\mathrm{SS}}| < \infty$: the "last SS" $\bar{\ell}_{\mathrm{SS}}$ occurs, after which only NS are performed, which means that $\bar{x}$ is fixed. The result hinges on estimating how much the insertion of new elements in $\mathcal{B}$ changes $\nu(2.14)$ if an NS is performed. Our version is quite similar to those that have been repeatedly used (cf., e.g., [1, 2]), except that it explicitly deals with the fact that only a subset of the components may have been evaluated. To simplify the notation, we temporarily drop the iteration index ℓ, and we denote with (2.14$_+$) and t_+, respectively, the master problem and the stabilization parameter at iteration $\ell + 1$. Crucial in our argument is the well-known aggregation technique, which inspired (3.7) for $\mathcal{P}$ and that is also central to lower bundle management (cf. Assumption 3). Again, the (dual) optimal solutions (2.16) can be used to define the *poorman's lower bundles*

$$\mathcal{B}^k_* = \{\, (z^k_*, \alpha^k_*) \,\} , \qquad k \in \mathcal{K} \;\; , \tag{3.9}$$

which have the property that if each $\mathcal{B}^k$ is replaced with the corresponding $\mathcal{B}^k_*$, and nothing else (i.e., $\bar{x}$ and t) is changed in the master problem, then the primal optimal solutions d_* and v^k_* remain the same, as (2.15) readily shows. In particular, this means that $\check{f}^k_{\mathcal{B}_*}(x_+) - \hat{f}^k_{\mathcal{P}}(\bar{x}) = \check{f}^k_{\mathcal{B}}(x_+) - \hat{f}^k_{\mathcal{P}}(\bar{x}) = d_* z^k_* - \alpha^k_* = v^k_*$. To proceed, as in section 3.3 we denote by $\mathcal{Z} \subseteq \mathcal{K}$ the set of the components for which a pair (z^k_+, α^k_+) has been added to $\mathcal{B}^k$. Since we examine the case of an NS, in the lemma we will assume that the condition

$$\Delta f^{\mathcal{Z}} := \sum_{k \in \mathcal{Z}} \big(\, \Delta f^k := \underline{f}^k_+ - \check{f}^k_+ \,\big) \geq m_1 \Delta_* \;\; \Leftrightarrow \;\; \underline{f}^{\mathcal{Z}}_+ \geq \check{f}^{\mathcal{Z}}_+ + m_1 \Delta_* \tag{3.10}$$

holds. It is easy to see that (3.10) is equivalent to (3.6), since for all $k \notin \mathcal{Z}$ one has $\underline{f}^k_+ = \check{f}^k_+$ (the lower model has not been improved, cf. (3.1)), i.e., $\Delta f^k = 0$; as a consequence, $\underline{f}_+ - \check{f}_+ = \Delta f^{\mathcal{Z}}$.

LEMMA 3.2. *If* (3.10), $\mathcal{B}^k_* \subseteq \mathcal{B}^k_+$ *for all* $k \in \mathcal{K}$, $\{(z^k, \alpha^k)\} \subseteq \mathcal{B}^k_+$ *for all* $k \in \mathcal{Z}$ *and* $t_+ \leq t$ *hold, then*

$$\nu(2.14) - \nu(2.14_+) \geq \frac{\Delta f^{\mathcal{Z}}}{2} \min \left\{ 1 \;,\; \frac{\Delta f^{\mathcal{Z}}}{t_+ \, \| z^{\mathcal{Z}}_* - z^{\mathcal{Z}} \|^2} \right\} \;\; . \tag{3.11}$$

Proof. We define the "minimal, aggregated" form of (2.14$_+$) as

$$\min \;\left\{\; \frac{1}{2} t \, \big\| z^{-\mathcal{Z}}_* + (1-\theta) z^{\mathcal{Z}}_* + \theta z^{\mathcal{Z}} \big\|^2 + \alpha^{-\mathcal{Z}}_* + (1-\theta)\alpha^{\mathcal{Z}}_* + \theta \alpha^{\mathcal{Z}} \;\; : \;\; \theta \in [0,1] \;\right\} \;\; . \tag{3.12}$$

Clearly, (3.12) is a *restriction* of (2.14$_+$) even under the minimal assumption that $\mathcal{B}^k_+ = \mathcal{B}^k_* \cup \{(z^k, \alpha^k)\}$ (cf. (3.9)). Indeed, (3.12) is the dual of (2.7) using the aggregated model (2.3) (cf. (3.19))—as opposed to the sum of individual models (2.4) as in (2.12)—with

$$\mathcal{B} = \big\{\, (z_*, \alpha_*) \,,\, (\bar{z}^{\mathcal{Z}}, \bar{\alpha}^{\mathcal{Z}}) \;=\; \big(\, z^{\mathcal{Z}} + z^{-\mathcal{Z}}_* \,,\, \alpha^{\mathcal{Z}} + \alpha^{-\mathcal{Z}}_* \,\big) \,\big\} \;\; . \tag{3.13}$$

In other words, (3.12) is a restriction of (2.14$_+$)—with "minimal" dual bundle—where $\theta^k = \theta$ for all $k \in \mathcal{K}$. Whichever way it is looked at, it is clear that, under Assumption 3, $\nu(3.12) \geq \nu(2.14_+)$ holds (using $t_+ \leq t$). Hence, we want to estimate

$$\nu(2.14) - \nu(2.14_+) \geq \zeta = \nu(2.14) - \nu(3.12) \geq 0 \;\; ,$$

where the last inequality comes from the fact that (z_*, α_*) is optimal for (2.14), and $\theta = 0$ is feasible in (3.12) and produces the same solution. Estimating ζ requires simple but tedious algebra, starting with

$$\begin{aligned}
\zeta =& \frac{t}{2} \|z_*\|^2 + \alpha_* - \min_{\theta \in [0,1]} \left\{ \frac{t}{2} \big\| z^{-\mathcal{Z}}_* + (1-\theta) z^{\mathcal{Z}}_* + \theta z^{\mathcal{Z}} \big\|^2 + \alpha^{-\mathcal{Z}}_* + (1-\theta)\alpha^{\mathcal{Z}}_* + \theta \alpha^{\mathcal{Z}} \right\} \\
=& \max_{\theta \in [0,1]} \left\{ \frac{t}{2} \|z_*\|^2 + \alpha_* - \frac{t}{2} \big\| z^{-\mathcal{Z}}_* + (1-\theta) z^{\mathcal{Z}}_* + \theta z^{\mathcal{Z}} \big\|^2 - \alpha^{-\mathcal{Z}}_* - (1-\theta)\alpha^{\mathcal{Z}}_* - \theta \alpha^{\mathcal{Z}} \right\}
\end{aligned}$$

and then using $z_* = z_*^{-\mathcal{Z}} + z_*^{\mathcal{Z}}$, $\alpha_* = \alpha_*^{-\mathcal{Z}} + \alpha_*^{\mathcal{Z}}$ to rewrite the objective function as

$$
\begin{aligned}
&\frac{t}{2}\left\| z_*^{-\mathcal{Z}} + z_*^{\mathcal{Z}} \right\|^2 - \frac{t}{2}\left\| z_*^{-\mathcal{Z}} + (1-\theta)z_*^{\mathcal{Z}} + \theta z^{\mathcal{Z}} \right\|^2 + \theta(\alpha_*^{\mathcal{Z}} - \alpha^{\mathcal{Z}}) = \\
&\quad = \frac{t}{2}\Big[\left\| z_*^{-\mathcal{Z}} \right\|^2 + 2z_*^{-\mathcal{Z}} z_*^{\mathcal{Z}} + \left\| z_*^{\mathcal{Z}} \right\|^2 - \left\| z_*^{-\mathcal{Z}} \right\|^2 - (1-\theta)^2 \left\| z_*^{\mathcal{Z}} \right\|^2 - \theta^2 \left\| z^{\mathcal{Z}} \right\|^2 \\
&\qquad\qquad - 2(1-\theta) z_*^{-\mathcal{Z}} z_*^{\mathcal{Z}} - 2\theta z_*^{-\mathcal{Z}} z^{\mathcal{Z}} - 2(1-\theta)\theta z_*^{\mathcal{Z}} z^{\mathcal{Z}} \Big] + \theta(\alpha_*^{\mathcal{Z}} - \alpha^{\mathcal{Z}}) \\
&\quad = -\frac{t}{2}\theta^2 \left\| z_*^{\mathcal{Z}} - z^{\mathcal{Z}} \right\|^2 + \theta\Big[t(z_*^{\mathcal{Z}} + z_*^{-\mathcal{Z}}) z_*^{\mathcal{Z}} - t(z_*^{-\mathcal{Z}} + z_*^{\mathcal{Z}}) z^{\mathcal{Z}} + \alpha_*^{\mathcal{Z}} - \alpha^{\mathcal{Z}} \Big] \\
&\quad = -\frac{t}{2}\theta^2 \left\| z_*^{\mathcal{Z}} - z^{\mathcal{Z}} \right\|^2 + \theta\big(- d_* z_*^{\mathcal{Z}} + d_* z^{\mathcal{Z}} + \alpha_*^{\mathcal{Z}} - \alpha^{\mathcal{Z}} \big),
\end{aligned}
$$

where in the last step we have used (2.15): $d_* = -t(z_* = z_*^{\mathcal{Z}} + z_*^{-\mathcal{Z}})$. By summing (3.3) and (2.19) over $k \in \mathcal{Z}$ we obtain, respectively, $\underline{f}_+^{\mathcal{Z}} - \hat{f}^{\mathcal{Z}} = z^{\mathcal{Z}} d_* - \alpha^{\mathcal{Z}}$ and $d_* z_*^{\mathcal{Z}} - \alpha_*^{\mathcal{Z}} = v_*^{\mathcal{Z}}$. Plugging these in the last line of the above derivation we finally conclude

$$
\zeta \geq \max\left\{ h(\theta) = \theta \Delta f^{\mathcal{Z}} - \tfrac{1}{2}\theta^2 M^{\mathcal{Z}} \ : \ \theta \in [0,1] \right\} \ , \tag{3.14}
$$

where $M^{\mathcal{Z}} = t \left\| z_*^{\mathcal{Z}} - z^{\mathcal{Z}} \right\|^2$ (≥ 0), and, by (3.10), $\Delta f^{\mathcal{Z}} = \underline{f}_+^{\mathcal{Z}} - \check{f}_+^{\mathcal{Z}}$ (≥ 0). Obviously, $\nu(3.14) \geq 0$ ($\theta = 0$ is feasible), showing that $\zeta \geq 0$. Since $h'(\theta) = \Delta f^{\mathcal{Z}} - M^{\mathcal{Z}}\theta$, if $M^{\mathcal{Z}} = 0 \Leftrightarrow z^{\mathcal{Z}} = z_*^{\mathcal{Z}}$, then h is linear and $h' = \Delta f^{\mathcal{Z}} \geq 0$: hence, the optimal solution to (3.14) is $\theta_* = 1$. Otherwise, the unconstrained maximum of h is $\bar{\theta} = \Delta f^{\mathcal{Z}} / M^{\mathcal{Z}} \geq 0$. If $\bar{\theta} \leq 1$, then $\theta_* = \bar{\theta}$ with $h(\theta_*) = h(\bar{\theta}) = (\Delta f^{\mathcal{Z}})^2 / (2M^{\mathcal{Z}})$. If, instead, $\bar{\theta} > 1$, then again $\theta_* = 1$, and therefore $h(\theta_*) = h(1) = \Delta f^{\mathcal{Z}} - M^{\mathcal{Z}}/2$. Hence,

$$
\begin{aligned}
&\Delta f^{\mathcal{Z}} \leq M^{\mathcal{Z}} \quad \Longrightarrow && && \zeta \geq (\Delta f^{\mathcal{Z}})^2/(2M^{\mathcal{Z}}) \ , \\
&\Delta f^{\mathcal{Z}} > M^{\mathcal{Z}} \quad \Longrightarrow && h(1) = \Delta f^{\mathcal{Z}} - \frac{1}{2}M^{\mathcal{Z}} > \frac{1}{2}\Delta f^{\mathcal{Z}} && \Longrightarrow \quad \zeta \geq \frac{1}{2}\Delta f^{\mathcal{Z}} \ .
\end{aligned}
$$

Furthermore $\Delta f^{\mathcal{Z}} \leq M^{\mathcal{Z}} \ \Leftrightarrow \ \Delta f^{\mathcal{Z}}/M^{\mathcal{Z}} \leq 1 \ \Leftrightarrow \ (\Delta f^{\mathcal{Z}})^2/(2M^{\mathcal{Z}}) \leq \frac{1}{2}\Delta f^{\mathcal{Z}}$, which finally allows us to conclude that

$$
\zeta \geq \nu(3.14) \geq \min\left\{ \frac{\Delta f^{\mathcal{Z}}}{2} \ , \ \frac{(\Delta f^{\mathcal{Z}})^2}{2M^{\mathcal{Z}}} \right\} = \frac{\Delta f^{\mathcal{Z}}}{2} \min\left\{ 1 \ , \ \frac{\Delta f^{\mathcal{Z}}}{t \left\| z_*^{\mathcal{Z}} - z^{\mathcal{Z}} \right\|^2} \right\} \ ,
$$

i.e., (3.11) holds. The formula also works when $z_*^{\mathcal{Z}} = z^{\mathcal{Z}} \Longrightarrow \theta_* = 1 \Longrightarrow \zeta \geq \Delta f^{\mathcal{Z}}$ by considering $\min\{ 1 \, , \, \Delta f^{\mathcal{Z}}/0 \} = \min\{ 1 \, , \, \infty \} = 1$. □

The lemma immediately suggests the analogue of Assumption 2 for $\mathcal{B}$. (Actually, as the following is well-known, it is Assumption 2 that has been suggested by analogy.)

Assumption 3 (lower bundle management).

(i) *For all $\ell > \bar{\ell}_{\mathsf{SS}}$ and $k \in \mathcal{K}$, let $\theta_{*,\ell}^k$ be the optimal solution to* (2.14)*: then, either all the $(z_i^k, \alpha_i^k) \in \mathcal{B}_\ell^k$ such that $\theta_{*,i,\ell}^k > 0$ also belong to $\mathcal{B}_{\ell+1}^k$, or $(\, z_*^k \, , \, \alpha_*^k \,) \in \mathcal{B}_{\ell+1}^k$ (cf.* (3.9)*).*

(ii) *There exists $\ell' \geq \bar{\ell}_{\mathsf{SS}}$ such that t_ℓ is nonincreasing for all $\ell \geq \ell'$.*

Assumption 3 is again written in an abstract form, in that it only has to hold *within the last infinite sequence of consecutive NS* (if any). Since it is impossible to say if $\ell > \bar{\ell}_{\mathsf{SS}}$, the conditions have to hold, eventually, within *any* sequence of *consecutive* NS. Yet, as soon as an SS is performed, each $\mathcal{B}^k$ can be entirely reset: hence, Assumption 3(i) is weaker than Assumption 2(i). Assumption 3(ii) allows online tuning of t_ℓ, which is well-known to be crucial in practice: it is not necessarily true that "the best" $t_{\ell+1}$ after an NS is smaller than t_ℓ (e.g., [1]). Of course, assuming that t_ℓ decreases whenever an NS is performed, as done, e.g., in [12, (6.14)], is a simple way to achieve the desired property. The combined effect of 3(i) and 3 (ii) is that, during a sequence of consecutive NS, eventually the values $\nu(2.14)$ are nonincreasing. Indeed, the previous optimal solution of (2.14) remains feasible (albeit, if (3.9) has been used, in a "surrogate form"), so that $\nu(2.14)$ cannot increase, since t_ℓ is also nonincreasing. This is the crux of the analysis of the case of an infinite sequence of consecutive NS.

THEOREM 3.3. *Under Assumptions* 1 *and* 3*, if* $|\mathcal{L}_{\mathsf{SS}}| < \infty$ *and* $|\mathcal{L}| = \infty$*, then* $\|z_{*,\ell}\| \to 0$ *and* $\alpha_{*,\ell} \to 0$*. Hence,* $\bar{x}$ *is optimal for* (1.1) *and* $\liminf_{\ell\to\infty} \hat{f}_\ell = \limsup_{\ell\to\infty} \check{f}_\ell = f(\bar{x}) = \nu(1.1)$.

Proof. We claim that $\Delta_{*,\ell} = \nu(2.14_\ell) \to 0$ as $\ell \to \infty$, which implies $\|z_{*,\ell}\| \to 0$ and $\alpha_{*,\ell} \to 0$. Once the claim is proven, the statement easily follows: (2.18) proves that $0 \in \partial f(\bar{x})$, and $\liminf_{\ell\to\infty} \hat{f}_\ell = \limsup_{\ell\to\infty} \check{f}_\ell = f(\bar{x}) = \nu(1.1)$—asymptotically, both the upper and the lower estimates become tight—is proved exactly as in Theorem 3.1.

By contradiction, assume $\Delta_{*,\ell} \geq \varepsilon > 0$ for all ℓ. Because NS are always performed, (3.6) (and therefore (3.10)) always holds: $\Delta f_\ell^{\mathcal{Z}} \geq m_1 \Delta_{*,\ell} \geq m_1\varepsilon > 0$. The hypotheses of Lemma 3.2 are satisfied, and therefore (3.11) holds for all ℓ ($\geq \ell'$). By Lemma 2.1, $\|z_i^k\| \leq L^k$; hence $z_{*,\ell}^k$ as a convex combination satisfies $\|z_{*,\ell}^k\| \leq L^k$ as well. Consequently, $\|z_{*,\ell}^{\mathcal{Z}} - z_\ell^{\mathcal{Z}}\|$ is bounded above by some constant, and since $t_{\ell+1} \leq t_\ell$, t_ℓ is also bounded above. The denominator of the rightmost term in (3.11) is also bounded above. Therefore, since $\Delta f_\ell^{\mathcal{Z}} \geq m_1\varepsilon > 0$, the whole rightmost term in (3.11) is bounded away from zero: hence, $\nu(2.14_{\ell+1}) < \nu(2.14_\ell) - \delta$ for some $\delta > 0$. But this means that $\nu(2.14_\ell) \to -\infty$ as $\ell \to \infty$, which contradicts $\nu(2.14) \geq 0$, thereby proving the claim and concluding the proof. $\square$

All in all, we reproduce the standard result.

THEOREM 3.4. *Under Assumptions* 1, 2, 3*, and* $\nu(1.1) > -\infty$*, if* $\delta_1 > 0$ *and* $\delta_2 > 0$*, then the algorithm finitely terminates with an* approximately *δ_2-optimal solution. If $\delta_1 = 0$, then $\{\,\bar{x}_\ell\,\}$ converges (possibly, finitely) to a δ_2-optimal solution of problem* (1.1).

Proof. Irrespectively of which among $|\mathcal{L}_{\mathsf{SS}}| = \infty$ and $|\mathcal{L}_{\mathsf{SS}}| < \infty$ happens, $\|z_{*,\ell}\| \to 0$ and $\alpha_{*,\ell} \to 0$ (by Theorems 3.1 and 3.3, respectively); hence, if $\delta_1 > 0$ and $\delta_2 > 0$, then the algorithm finitely terminates. When this happens, $z_{*,\ell} \in \partial_{\alpha_{*,\ell}} f(\bar{x}_\ell)$ (cf. (2.18)), which, since $\|z_{*,\ell}\|$ is "small," is an appropriate notion of approximate δ_2-optimality (as $\alpha_{*,\ell} \leq \delta_2$). If $\delta_1 = 0$ and the algorithm terminates, then $\bar{x}_\ell$ is δ_2-optimal. Otherwise Theorems 3.1 and 3.3 give the desired result. $\square$

As the proof shows, if it ever happens that $z_{*,\ell} = 0$, then the algorithm provides a correct *a posteriori* estimate of the error: $\bar{x}_\ell$ is, then, $\alpha_{*,\ell}$-optimal. This is specific of our setting, and due to the fact that the oracles produce upper estimates, whose gap with the lower ones is "incorporated" in the α_i^k.

3.3. The inner loop.

The goal of the inner loop is to attain Assumption 1 "with the least possible computational effort." This means calling as few oracles as possible with the loosest possible parameters (larger upper target and accuracy, smaller lower target). A general scheme of inner loop is as follows:

0 (Input and initialization) Input initial lower/upper bundles $\mathcal{B}/\mathcal{P}$, the stability center $\bar{x}$ and iterate x_+ to be evaluated, global lower/upper targets $\underline{\mathtt{tar}}/\overline{\mathtt{tar}}$ and accuracy ε, disaggregated values v^k_* for all $k \in \mathcal{K}$, the aggregated value Δ_*, parameters $0 < m_1 < m_2 < 1$. Set $\mathcal{Z} \hookleftarrow \emptyset$. For each $k \in \mathcal{K}$, evaluate (2.6) at x_+ and add $(x_+, \hat{f}^k_+)$ to $\mathcal{P}^k$. With arbitrary weights $\beta^k \geq 0$ such that $\sum_{k \in \mathcal{K}} \beta^k = 1$, define the individual upper and lower targets

$$\check{f}^k_+ = \hat{f}^k + v^k_* \quad < \quad \underline{\mathtt{tar}}^k \hookleftarrow \check{f}^k_+ + m_1 \beta^k \Delta_* \quad < \quad \overline{\mathtt{tar}}^k \hookleftarrow \check{f}^k_+ - m_2 \beta^k v_* \ . \tag{3.15}$$

1 (Oracle call) Arbitrarily select $k \in \mathcal{K}$ and the accuracy $\varepsilon^k \geq \beta^k \varepsilon$. Call $\mathcal{O}^k(\,\min\{\,\underline{\mathtt{tar}}^k\,,\,\underline{\mathtt{tar}} - \underline{f}^{-k}_+\,\}, \max\{\,\overline{\mathtt{tar}}^k\,,\,\overline{\mathtt{tar}} - \bar{f}^{-k}_+\,\}\ ,\ \varepsilon^k\ ,\ x_+\,)$, collect the output $\underline{f}^k_+$, $\bar{f}^k_+$ and possibly z^k. Update $\underline{f}^k_+$ and $\bar{f}^k_+$ according to (3.1).

2 (Bundle management) Add $(x_+, \bar{f}^k_+)$ to $\mathcal{P}^k$ replacing the previous pair. `If` $\underline{f}^k_+ > -\infty$ held before the update (3.1), and therefore z^k has been produced, `then` $\mathcal{Z} \hookleftarrow \mathcal{Z} \cup \{k\}$ and add $(z^k, \alpha^k_+(\bar{x}, \mathcal{P}))$ to $\mathcal{B}^k$.

3 (Termination) `If` neither (3.5) nor (3.6) hold, `then go to` Step 1.

4 (Output) Output updated lower/upper bundles $\mathcal{B}/\mathcal{P}$.

We now discuss the rationale of the specific choice of the oracle targets. First of all, that the targets are defined as in "$\min\{\,\underline{\mathtt{tar}}^k\,,\,\underline{\mathtt{tar}} - \underline{f}^{-k}_+\,\}$" depends on the fact that one of the two terms corresponds to being able to immediately declare an SS/NS. For instance, if $\underline{\mathtt{tar}}^k > \underline{\mathtt{tar}} - \underline{f}^{-k}_+$, and $\underline{f}^k_+ \geq \underline{\mathtt{tar}} - \underline{f}^{-k}_+$, then (3.6) immediately holds and the inner loop can immediately end, and similarly for the upper target. We need only then discuss the case where the oracle targets are actually defined by (3.15). Then, due to (2.2)(iii), after the call to $\mathcal{O}^k$ *at least* one of the "partial" SS/NS conditions

$$\text{(i)} \quad \hat{f}^k_+ \leq \overline{\mathtt{tar}}^k \qquad , \qquad \text{(ii)} \quad \check{f}^k_+ \geq \underline{\mathtt{tar}}^k \tag{3.16}$$

holds. By summing over k, one establishes that if (3.16)(i) holds for *all* $k \in \mathcal{K}$, then (3.5) holds, and, similarly, if (3.16)(ii) holds for *all* $k \in \mathcal{K}$, then (3.6) holds. Hence, initially calling $\mathcal{O}^k$ with a "large" ε^k, say, $\varepsilon^k = \infty$, makes sense: if the oracles of *all* components satisfy *the same* relation in (3.16), then either an SS or an NS is done, possibly without having ever produced a single lower/upper estimate at that iterate. Actually, due to the use of the upper/lower models an SS/NS can be performed even (quite) before having computed all components. However, this may not happen; for instance, some oracle may return an upper estimate but no lower one, and some may do the converse. Thus, ultimately it may be necessary to call the oracles with "small enough" accuracy, in order to ensure that one among (3.5) and (3.6) holds. It is easy to verify that such a small enough accuracy exists. Indeed, calling $\mathcal{O}^k$ with $\varepsilon^k = \beta^k \varepsilon$ for all $k \in \mathcal{K}$ clearly suffices, because then (2.2)(i) gives $\hat{f}^k_+ - \check{f}^k_+ \leq \beta^k \varepsilon$ for each $k \in \mathcal{K}$, and therefore $\hat{f}_+ - \check{f}_+ \leq \varepsilon$. If *both* (3.5) and (3.6) were not satisfied one would have

$$\hat{f}_+ - \check{f}_+ > \overline{\mathtt{tar}} - \underline{\mathtt{tar}} = -m_2 v_* - m_1 \Delta_* = \varepsilon \ , \tag{3.17}$$

i.e., a contradiction. This immediately suggests the following.

Assumption 4 (accuracy management in the inner loop). *After a finite number of inner iterations, for all $k \in \mathcal{K}$ the oracle $\mathcal{O}^k$ has been called with $\varepsilon^k = \beta^k \varepsilon$.*

By the above discussion, it is clear that Assumption 4 readily implies Assumption 1. There are many possible ways in which this can be obtained. It is sensible to start by setting $\varepsilon^k = \infty$ for all $k \in \mathcal{K}$, so as to give the oracles the chance to terminate without providing one of the two bounds. If this fails to attain either (3.5) or (3.6), then the oracles can be called again with a smaller ε^k, up until $\varepsilon^k = \beta^k \varepsilon$. The exact sequence of the values ε^k for a given k, as well as the exact order in which the different components are evaluated, is immaterial for the convergence (although it can, of course, have a significant influence on the actual computational cost). It is also appropriate to remark that calling the same oracle $\mathcal{O}^k$ more than once with the same iterate x_+ but decreasing ε^k also makes computational sense: in many cases the oracle will *reoptimize efficiently* at little or no implementation cost. Typically, state-of-the-art optimization solvers allow one to terminate the search for an optimal solution early by specifying a coarse optimality tolerance, and then resume the computation of a more accurate solution at little or no extra cost by keeping all their internal data structures updated and just "jumping into the main loop." This is, of course, not the only possible way in which reoptimization can be done: more sophisticated hot-starting approaches can also be designed if $x_{\ell+1}$ was already processed several iterations ago (e.g., the Lagrangian dual hot-starting procedure described in [36]). However, doing reoptimization by calling the oracle multiple times on the same x_+ is, in many applications, basically "free." Furthermore, while reoptimizing linear/convex problems is well-understood, doing the same with, say, integer programs can be an issue due to the large size of the internal state of the algorithm that has to be kept (the whole Branch & Bound tree). So, for several applications this form of reoptimization may be the only practical one.

It is also appropriate to remark about the role of the somewhat "artificial-looking" combinators β^k. In view of the standard analysis of bundle methods, it would have been more natural to define $\overline{\mathtt{tar}}^k$ using v_*^k rather than $\beta^k v_*$. Defining β^k might have been necessary anyway to split the aggregated quantity Δ_* across the components in order to define $\underline{\mathtt{tar}}^k$, but it would most definitely have been natural to assume $\beta^k v_* = v_*^k$. This is, however, not possible because $v_*^k > 0$ may happen for some (but not for all) $k \in \mathcal{K}$: using v_*^k in place of $\beta^k v_*$ might then lead to $\overline{\mathtt{tar}}^k < \underline{\mathtt{tar}}^k$. In the same vein, note that in (3.15) we have *not* indicated that $\overline{\mathtt{tar}}^k < \hat{f}_+^k$, precisely because this may fail to be true, in particular when $v_*^k > 0$. Indeed, this is the case of a component that is actually increasing along d_*, but its increase is counterbalanced by the others. For such a component it may even happen that $\hat{f}_+^k = \check{f}_+^k$, so that $\overline{\mathtt{tar}}^k > \hat{f}_+^k$. This is not an issue, though; on the contrary, it says that such a component may be basically irrelevant in the choice between an SS and an NS, as the others alone may be enough to dictate it. For instance, if (3.16)(i) would happen to hold for all other components, then (3.5) would surely hold even without even calling $\mathcal{O}^k$. This is a nice illustration of the fact that the upper model can allow to perform an SS without having called all oracles on x_+, which is precisely the aim of the development.

Finally, we remark that if $\delta_2 > 0$, then $\varepsilon = -m_2 v_* - m_1 \Delta_* \geq (m_2 - m_1)\alpha_* \geq (m_2 - m_1)\delta_2$. Hence, no oracle will ever be required to reply with an *absolute* accuracy greater than (approximately, say, if $m_1 \approx 1$ and $m_2 \approx 0$) some factor β^k of the total *absolute* accuracy required for the solution of (1.1). By having β^k somehow related to f_+^k / f_+ (cf., e.g., section 3.6) we can have each oracle to never be required to yield a solution with (approximately) more than the *relative* accuracy required for the solution of (1.1), which is the best that one could expect [9, Observation 2.7]. Yet, during the course of the algorithm, when $\alpha_* \gg \delta_2$, the required accuracy will automatically be (much) less than that; and even that accuracy will only be required when it is strictly necessary to allow the algorithm to proceed.

3.4. Extension: The (partly) aggregated case. Using a *disaggregated approach*—with model (2.4), and therefore master problems (2.12)/(2.14)—is natural in our setting. This has been reported to be beneficial in terms of convergence speed (e.g., [4, 19, 26]), but the increased size of the master problem can eventually dominate overall running time (e.g., [19]). In some cases [38], it might therefore be beneficial to rather use the *aggregated approach*, with the *aggregated* bundle $\mathcal{B}$ and master problems

$$\min \left\{ v + \frac{1}{2t} \|d\|^2 \ : \ v \geq z_i d - \alpha_i \quad i \in \mathcal{B} \right\}, \tag{3.18}$$

$$\min \ \left\{ \frac{1}{2}t \left\| \sum_{i \in \mathcal{B}} z_i \theta_i \right\|^2 + \sum_{i \in \mathcal{B}} \alpha_i \theta_i \ : \ \theta \in \Theta \right\} \tag{3.19}$$

solved instead of (2.12)/(2.14). The fact that one has to sum over *all* $k \in \mathcal{K}$ seems to fly squarely in the face of an incremental method, where at each iteration one is trying to compute as few components as possible. Yet there are at least two possible strategies to extend the approach to an aggregated setting.

Full aggregation. The main idea is to keep individual disaggregate bundle (both lower and upper) information despite the use of (3.18)/(3.19). We thus do the following:

- Keep *the disaggregated representation of each z_i and α_i in terms of their individual components z_i^k and α_i^k.* This allows us to reconstruct the individual v_*^k, z_*^k, and α_*^k by just using the unique solution θ_* of (2.14) uniformly for all $k \in \mathcal{K}$.
- Keep the disaggregate upper bundle $\mathcal{P}^k$, which for the "poorman's" (3.7) with $x_* = \bar{x}_\ell$ only amounts to keeping the separate values $\bar{f}^k$ together with the aggregated one $\bar{f}$.

Then, since the SS condition (3.5) only depends on the aggregated v_*, Theorem 3.1 still holds true. The individual $\hat{f}^k$ and v_*^k are required to form the targets (3.15) and therefore the accuracy ε^k of each $k \in \mathcal{K}$, but as soon as (3.5) is obtained an SS can be performed. Forming a unique (z_+, α_+) even if (3.6) is triggered with $\mathcal{Z} \subsetneq \mathcal{K}$ is also easy: one just has to use $(\bar{z}^{\mathcal{Z}}, \bar{\alpha}^{\mathcal{Z}})$ of (3.13). Indeed, this is all that is needed to form the "approximated" master problem (3.12), which is the crux of Lemma 3.2 providing the crucial estimate (3.11), and therefore Theorem 3.3 still works with the obvious modification to Assumption 3(i).

Partial aggregation. Keeping the disaggregated representation (z_i^k, α_i^k) for all $k \in \mathcal{K}$ of each (z_i, α_i) may come at a considerable memory cost, and the computational cost of forming the individual v_*^k, z_*^k and α_*^k may be nonnegligible, too. This can actually be avoided by an alternative approach: insert *partly aggregated cuts/variables*

$$\sum_{k \in \mathcal{Z}} v^k \geq z_i^{\mathcal{Z}} d - \alpha_i^{\mathcal{Z}} \tag{3.20}$$

in the primal/dual disaggregated master problems (2.12)/(2.14). It is easy to see that using (3.20) does not impair the fundamental property of the master problem, i.e., that $v_*^k \leq f^k(\bar{x} + d_*)$: indeed, (3.20) is the *surrogate constraint* of the $|\mathcal{Z}|$ constraints $v^k \geq z_i^k d - \alpha_i^k$, and therefore (2.12) with (3.20) is a relaxation of (2.12) with these. In the dual, the variable $\theta_i^{\mathcal{Z}}$ associated with $(z_i^{\mathcal{Z}}, \alpha_i^{\mathcal{Z}})$ participates to the simplex constraints for all (and only) the $k \in \mathcal{Z}$. Actually, nothing prevents arbitrarily partitioning $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2 \cup \ldots \cup \mathcal{Z}_p$, and inserting the corresponding p partly aggregated cuts in (2.14): $p = |\mathcal{Z}|$ reproduces the disaggregated approach. Hence, the trade-off between master problem size/cost and convergence speed can be explored, ranging from fully aggregated models, to fully disaggregated ones, to "anything in between." Choosing the best aggregation level can only be done computationally; alternatively, along the lines of [30, 34] one may consider versions wherein the partition automatically adapts. A static a priori regrouping has been shown to be beneficial in [23]. Dynamically regrouping components is the subject of the recent work [22] and better left to future study.

3.5. Extension: Constraints/easy components. We now discuss dealing with constraints $x \in X$ in (1.1). Actually, in the spirit of [19] we will treat a more general case: that where f has components $\mathcal{K} = \{\, 0 \,\} \cup \mathcal{K}''$, where f^0 is "easy" in the sense that it can be *effectively written into the master problem.* This covers different situations, such as

1. $f^0 = \mathbb{1}_X$ with X represented by "few, simple" (say, conic) constraints;
2. $f^0(x) = bx$ with b the right-hand side of the relaxed constraints in (1.4);
3. $f^0 = \nu(1.3^0_x)$, where U^0 can be represented with "few, simple" constraints [19].

In all these cases (and the combinations thereof), the natural way to deal with f^0 is just to insert it unmodified in the master problem, which then becomes

$$\min\left\{\; f^0(\bar{x}+d) + \sum_{k\in\mathcal{K}} v^k + \frac{1}{2t}\,\|d\|^2 \;:\; v^k \geq z^k_i d - \alpha^k_i \quad i \in \mathcal{B}^k \;\;,\;\; k \in \mathcal{K}'' \;\right\} . \tag{3.21}$$

For instance, when $f^0 = \mathbb{1}_X$ this amounts to adding the constraints "$\bar{x} + d \in X$" to (2.12), which ensures that $x_+ = \bar{x} + d_*$ is feasible; this may reasonably not make (3.21) much more costly to solve, e.g., when X is polyhedral. The *dual* of (3.21) is (see, e.g., [19])

$$\begin{cases} \min \;\; \frac{1}{2}t\,\left\| z^0 + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{B}^k} z^k_i\theta^k_i \right\|^2 + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{B}^k} \alpha^k_i\theta^k_i - \bar{x}z^0 + (f^0)^*(z^0), \\ \text{s.t.} \quad\; \theta^k \in \Theta^k, \quad k \in \mathcal{K}'', \end{cases} \tag{3.22}$$

where $(f^0)^*(z^0) = \sup_x\{\, xz^0 - f^0(x) \,\}$ is *Fenchel's conjugate* of f^0 (e.g., [29]). It is not difficult to check that, by employing (3.21)/(3.22) in place of (2.12)/(2.14) in the algorithm, together with some minor changes, everything can be made to work:

- Since function f^0 in (3.21) is not translated in value, the global predicted descent now is $v_* = f^0(\bar{x} + d_*) - f^0(\bar{x}) + \sum_{k\in\mathcal{K}} v^k_*$, where $f^0(\bar{x} + d_*)$ is *exactly* known as a by-product of solving (3.21). There is a minor difficulty for $\ell = 1$ as $f^0(\bar{x}_1)$ is not known, but this is not hard to circumvent: just assume $f^0(\bar{x}_1) = \infty$, forcing an SS at the first iteration.
- In the inner loop there is no need to call any oracle for f^0, nor is there any need to ever compute (2.6) for $k = 0$, as the exact value of f^0 is always known; note that for $f^0 = \mathbb{1}_X$ the value is always 0. Analogously, there is no need to compute any z^0 for f^0, as there is no $\mathcal{B}^0$ to be filled: the full description of f^0 is already present in (3.21), although in a different form.
- The individual targets (3.15) need not be defined for f^0, but f^0 does contribute to setting the oracle targets for $k \in \mathcal{K}''$, as $f^0(x_+)$ (the *exact* value) is counted in $\hat{f}^{-k}_+$. This only applies to $\overline{\texttt{tar}}^k$, though: since $\hat{f}^0_+ = \check{f}^0_+$, the 0th component will never contribute to an NS (cf. (3.10)), whose only effect is to improve the lower model $\check{f}_{\mathcal{B}}$, which cannot happen to an easy component. Note that for $f^0 = \mathbb{1}_X$ one always has $\bar{f}^0 = 0$, and hence the 0th component will not contribute to (3.5) either.

- Solving the (3.21) means also solving its dual (3.22), which produces a z^0_*, albeit not as a convex combination of z^0_i (typically, as dual optimal solutions of some constraints [19]). Similarly, it is possible to define a correct α^0_*. However, there is no use for (z^0_*, α^0_*): it does not have to be added to $\mathcal{B}^0$, and v^0_* is not computed through it. Yet, one has to be aware that the constant "$+f^0(\bar{x})$" has to be added to $\nu(3.22)$ in order for $\Delta^k = \nu(3.22) + f^0(\bar{x})$ to behave precisely as $\nu(2.14)$ ($\Delta^k = 0 \iff \bar{x}$ optimal).

Ultimately, it is easy to see that the convergence analysis of section 3.2 still applies. This also extends, mutatis mutandis, to the case of section 3.4: the (partly) aggregated master problems (3.18)/(3.19) can be modified similarly to (3.21)/(3.22) with the explicit term for f^0. That is, the $k \in \mathcal{K}''$ can be aggregated while f^0 remains disaggregated from them. This can have a surprisingly large effect on the convergence speed [19].

3.6. Implementations details. We now comment on some details of the algorithm.

- At Step 0, $\varepsilon^k_1 < \infty$ is arbitrary: $\mathcal{O}^k$ have to provide finite but upper and lower estimates, together with initial linearizations z^k_1, so that $\hat{f}_{\mathcal{P}} < \infty$ and $\check{f}_{\mathcal{B}} > -\infty$. Since estimates can be *arbitrarily loose*, this should be "cheap." It would actually be possible to call $\mathcal{O}^k$ at a different $\bar{x}^k_1$ for each $k \in K$ and arrange $\mathcal{B}^k_1/\mathcal{P}^k_1$ accordingly: (3.1) would still give finite estimates at $\bar{x}_1$.
- Setting $m_1 \approx 0$ and $m_2 \approx 1$ should clearly make the oracle cheaper. However, it also means that the decrease of $\hat{f}$ in an SS/of $\nu(2.14)$ in an NS can be "small," thereby increasing the iterations count. Thus, a nontrivial trade-off may have to be explored here.
- In Step 3 of the inner loop, one may elect not to terminate even if one among (3.5) and (3.6) holds, in order to attain better upper/lower estimates on f at x_+; these may improve convergence speed; cf. the previous point. Again, the trade-off here may be nontrivial. Clearly, doing this cannot impair convergence, provided that sooner or later the inner loop is terminated.
- The convex combinators β^k in the inner loop serve to "subdivide" the "desired amount of increase $m_1\Delta_{*,\ell}$/decrease $(1 - m_2)v_{*,\ell}$" (cf. Figure 1), which are defined for the whole of f, among the components. Since this is used to make targets for the oracles, it has to be done a priori: hence, some sort of "guess" about the individual values $f^k(x_{\ell+1})$ is unavoidable. The selection of the β^k may be significant computationally. Intuitively, it should take into account factors such as "how hard is each component to evaluate" ("easier" ones might get smaller β^k, as computing them with high accuracy is less demanding) and that some components may have significantly larger values than others, thereby being more influential on the overall value of f. A simple formula taking into account the latter factor (but not the former) is

$$\gamma^k_\ell = |v^k_{*,\ell}| / \left|\bar{f}^k_\ell\right| + 1 \quad , \quad \beta^k_\ell = \gamma^k_\ell / \sum_{h \in \mathcal{K}} \gamma^h_\ell \ .$$

- One may add checks to speed-up the algorithm if $\Delta\hat{f}_\ell > 0$, i.e., $\hat{f}_{\ell+1}(\bar{x}_\ell) < \hat{f}_\ell(\bar{x}_\ell)$ due to new information inserted in $\mathcal{P}$. For instance, since $\alpha_{*,\ell}$ decreases, one may recheck the stopping condition. Also, one may check if $\hat{f}_\ell$ has decreased by at least $(1 - m_2)v_{*,\ell}$ (cf. Theorem 3.1), and in case perform an SS/NS: set $\bar{x}_{\ell+1} = \bar{x}_\ell$ but declare this as an SS (which, in particular, means that $\mathcal{B}$ may be entirely cleared; cf. Assumption 3(i)). Similarly, one may check if $\Delta\hat{f}_\ell \geq \delta$ for some fixed $\delta > 0$: this means that $\nu(2.14)$ has decreased by at least δ (cf. Theorem 3.3), allowing one to declare an NS.

4. Uncooperative oracles and noise reduction. The treatment so far has hinged on a "gentleman's agreement" between the algorithm and the oracle: while the former always tries to ask as little accuracy as possible, providing the oracle with all possible clues about when it is possible to stop the computation without having attained an exact solution, the latter obliges itself to find a solution as accurate as required. Some oracles may find it very time consuming, or even impossible, to do that. It is therefore important to consider the case of a *uncooperative* oracles that may not be capable, or willing, to satisfy (2.2) with all possible values of $\overline{\mathtt{tar}}^k$, $\underline{\mathtt{tar}}^k$, and ε^k. We will analyze three types of uncooperative oracles, corresponding to different assumptions on the solution approaches available for (1.3_x^k). The overarching assumption will be that $\underline{f}^k$ and z^k at the very least satisfy (2.1), i.e., the oracle is *lower*: using nonlower oracles is also possible, as thoroughly analyzed in [12], but lower oracles are the natural approach for our target setting where subgradients are associated to feasible solutions of (1.3_x^k). Common to all cases is the fact that oracles will have to cooperate at least partly, which will mean that there will have to be a bound on the maximal error; this is why we term them *boundedly* uncooperative. While different assumptions on how the error may "reveal" itself lead to significantly different algorithmic schemes, all schemes will exploit the fact that bundle methods have a simple way to deal with uncooperative oracles: the fact that t is almost a "free parameter." Hence, if $\mathcal{O}$ is uncooperative at x_+, one may just try to generate a different iterate by changing t and hope that the oracle will, for whatever reason, be cooperative there. Any such mechanism requires some safeguard to avoid sampling X forever with the oracles stubbornly refusing to provide any valuable information. As we shall see, the safeguard is that, eventually, the stability parameter grows, which is called noise reduction. Although the theory is developed with the incremental case in mind, there is no explicit reference to the inner loop workings. In other words, it applies immediately to the nonincremental case (say, only one oracle) with informative but uncooperative oracles.

4.1. Informative boundedly uncooperative oracles. The first kind of oracle corresponds to solution methods for (1.3_x^k) that can provide upper and lower bounds, but cannot (or are not willing to) guarantee that these will be *arbitrarily* close, just "close up to a point." However, these (boundedly) uncooperative oracles are at least *informative* in the sense that they *explicitly declare* a priori the smallest accuracy value $0 < \bar{\varepsilon}^k < \infty$ that they can achieve. The typical example is *approximation methods with worst-case a priori guarantees* for (1.3_x^k). Hence, an informative uncooperative oracle will be able to satisfy (2.2) *only provided that* $\varepsilon^k \geq \bar{\varepsilon}^k$. When called with $\varepsilon^k < \bar{\varepsilon}^k$ it can only guarantee that

$$0 \leq f^k(x) - \underline{f}^k \leq \bar{f}^k - \underline{f}^k \leq \bar{\varepsilon}^k \quad \Longrightarrow \quad z^k \in \partial_{\bar{\varepsilon}^k} f^k(x) \ . \tag{4.1}$$

Hence, in particular $\mathcal{O}^k$ may not be able to satisfy (2.2)(iii) if $\overline{\mathtt{tar}}^k - \underline{\mathtt{tar}}^k < \bar{\varepsilon}^k$. Obviously, a collection of informative (boundedly) uncooperative oracles for all $k \in \mathcal{K}$ is an informative (boundedly) uncooperative oracle for f with $\bar{\varepsilon} = \bar{\varepsilon}^{\mathcal{K}} < \infty$. The issue with (4.1) is that Assumption 4 no longer yields the crucial Assumption 1: even after that all $\mathcal{O}^k$ have been called with $\varepsilon^k = \beta^k \varepsilon$, it is not guaranteed that one among (3.5) and (3.6) holds. To avoid this, one must avoid "asking too much" from each oracle. This is actually easy to obtain by including NR step in the main loop:

Step 2.1 (Noise reduction) `if` $\varepsilon_\ell < \bar{\varepsilon}$, `then if` $\|z_{*,\ell}\|^2 \leq \delta_1$, `then stop else` select $t_{\ell+1}$ appropriately; $\bar{x}_{\ell+1} \hookleftarrow \bar{x}_\ell$; $\mathcal{B}_{\ell+1} = \mathcal{B}_\ell$; $\mathcal{P}_{\ell+1} = \mathcal{P}_\ell$; $\ell \hookleftarrow \ell + 1$; `go to` Step 1;

The aim of Step 2.1 is to obtain $\varepsilon_{\ell+1} \geq \bar{\varepsilon}$; since $\Delta_{*,\ell} \leq -v_{*,\ell}$,

$$\varepsilon_\ell = \overline{\mathtt{tar}}_\ell - \underline{\mathtt{tar}}_\ell = m_2(-v_{*,\ell}) - m_1\Delta_{*,\ell} \geq (m_2 - m_1)(-v_{*,\ell}) \quad , \tag{4.2}$$

i.e., this can be obtained by increasing t_ℓ, which (hopefully) increases $v_{*,\ell}$ (cf. (2.15)). An estimate on how much t_ℓ has to be increased can be obtained by means of sensitivity analysis techniques on (2.12)/(2.14) [16, section 7]. Step 2.1 implies that whenever the oracles are called, it is possible for them to provide the expected answer; this also relies on the inner loop to choose reasonable values for β^k.

Assumption 5. *Assumption* 4 *holds with* β^k *chosen so that* $\varepsilon^k = \beta^k\varepsilon \geq \bar{\varepsilon}^k$ *for all* $k \in \mathcal{K}$.

In the rest of this section we will assume that Step 2.1 is in effect, which means that we can also impose Assumption 5, which in turn yields Assumption 1. Stopping in Step 2.1 is justified by the fact that $\bar{x}_\ell$ is then already "as optimal as it gets:"

Lemma 4.1. *If the algorithm stops in Step* 2.1*, then* $\bar{x}_\ell$ *is* approximately $\bar{\varepsilon}/(m_2 - m_1)$*-optimal.*

Proof. From $\bar{\varepsilon} > \varepsilon_\ell$, (4.2) and $\alpha_{*,\ell} \leq -v_{*,\ell}$ one has

$$\varepsilon' := \bar{\varepsilon}/(m_2 - m_1) \geq -v_{*,\ell} \geq \alpha_{*,\ell} \quad , \tag{4.3}$$

which, using (2.18) and the fact that $\|z_{*,\ell}\|^2 \leq \delta_1$, shows that $\bar{x}_\ell$ is *approximately* ε'-optimal. □

Hence, if the algorithm stops in Step 2.1, then $\bar{x}_\ell$ has (about) the best error that can be obtained given the oracle [9, Observation 2.7]: when $m_2 \approx 1$ and $m_1 \approx 0$, $\varepsilon' \approx \bar{\varepsilon}$. Ensuring that the stopping condition is eventually achieved basically only requires that eventually t_ℓ grows. With $\mathcal{L}_{\mathtt{NR}}$ the set of indices of NR, a simple and general way to state the necessary requirement is the following.

Assumption 6. *If* $|\mathcal{L}_{\mathtt{NR}}| = \infty$*, then* $\liminf_{\ell \ni \mathcal{L}_{\mathtt{NR}} \to \infty} t_\ell = \infty$.

Management of t_ℓ is therefore simple: if infinitely many NR are done, eventually $t_\ell \to \infty$ has to happen. This of course contrasts with Assumption 3(ii), but the latter is only there for the case of an infinite sequence of consecutive NS; as we shall see, this will not be an issue. An elementary scheme to attain Assumption 6 is based on a global memory $t_{\max}$; each time an NR is performed $t_{\ell+1} \hookleftarrow 2\max\{\, t_\ell \,,\, t_{\max} \,\}$ and $t_{\max} \hookleftarrow t_{\ell+1}$. Doing so guarantees that infinite sequences of (nonconsecutive) NR attain the same result as infinite sequences of (consecutive) NS.

Lemma 4.2. *Under Assumption* 6*,* $|\mathcal{L}_{\mathtt{NR}}| = \infty \Longrightarrow \|z_{*,\ell}\|^2 \to 0$.

Proof. From (4.2) and $-v_{*,\ell} \geq t_\ell \|z_{*,\ell}\|^2$ (cf. (2.15)) one has, using (4.3) for the second inequality, $\infty > \bar{\varepsilon} > (m_2 - m_1)t_\ell \|z_{*,\ell}\|^2$; $\|z_{*,\ell}\|^2 \geq \delta > 0$ would contradict $t_\ell \to \infty$. □

We remark that Lemma 4.1 together with Lemma 4.2 nicely illustrates the moniker "noise reduction." The oracle frames the "signal" $f^k(x_{\ell+1})$ between the two measures $\bar{f}^k_{\ell+1}$ and $\underline{f}^k_{\ell+1}$, affected by "noise" up to $\bar{\varepsilon}^k$. If $\varepsilon_\ell = \overline{\mathtt{tar}}^k_\ell - \underline{\mathtt{tar}}^k_\ell < \bar{\varepsilon}^k$, then the noise is too large to ensure that at least one of the two targets is achieved. Increasing

t_ℓ either increases $-v_{*,\ell}$, and therefore pulls further apart the targets, or decreases $\|z_{*,\ell}\|^2$. Hence, either the signal becomes significant w.r.t. the noise, or one detects (approximate) optimality. Although the idea of increasing t_ℓ when the algorithm gets stuck was originally proposed in [27], the mechanism to declare an NR there is significantly different from that in Step 2.1. Having t_ℓ grow ensures that, if $\delta_1 > 0$, NR cannot happen forever: eventually, either the algorithm stops in Step 2.1, or $-v_{*,\ell}$ grows large enough so that $\varepsilon_\ell < \bar{\varepsilon}$ no longer holds and a "regular" step is performed, yielding at least one among (3.5) and (3.6) due to Assumption 5. Thus the algorithm is well-defined, and the convergence theory of section 3.2 in principle applies again. However, in this setting infinitely many SS are ruled out, unless $\nu(1.1) = -\infty$.

LEMMA 4.3. *Either* $\hat{f}_\ell \to -\infty$ *or* $|\mathcal{L}_{\mathtt{SS}}| < \infty$.

Proof. Observe that $\Delta_{*,\ell} \geq -v_{*,\ell}/2$, and therefore $\varepsilon_\ell \leq (m_2 - m_1/2)(-v_{*,\ell})$. Thus, $\varepsilon_\ell \geq \bar{\varepsilon} > 0$ implies that $v_{*,\ell}$ is bounded away from 0, and (3.5) then gives that $\hat{f}_\ell \to -\infty$ if $|\mathcal{L}_{\mathtt{SS}}| = \infty$. □

As a consequence, it would be quite reasonable to reset $t_{\max}$ to arbitrary values at each SS: as Lemma 4.3 shows, there cannot be infinitely many of them unless the problem is being proved unbounded below. After the last SS, the problem has in fact been solved (to the highest possible accuracy).

THEOREM 4.4. *Under Assumption* 6*, if* $|\mathcal{L}| = \infty$ *and* $\hat{f}_\infty > -\infty$*, then* $\liminf_{\ell\to\infty} \alpha_{*,\ell} = \alpha_{*,\infty} \leq \varepsilon'$ *and the stability center* $\bar{x}$ *is* $\alpha_{*,\infty}$*-optimal. Here* $\hat{f}_\infty$ *is defined as* $\hat{f}_\infty = \lim_{\mathcal{L}_{\mathtt{SS}}\ni\ell\to\infty} \hat{f}_\ell$*.*

Proof. From Lemma 4.3, since $\hat{f}_\infty > -\infty$ then only finitely many SS are done. Hence, if also $|\mathcal{L}_{\mathtt{NR}}| < \infty$, then a tail of infinitely many consecutive NS starts: as in Theorem 3.3, $\|z_{*,\ell}\|^2 \to 0$ and $\alpha_{*,\ell} \to 0$, and hence $\bar{x}$ is optimal (and therefore a fortiori ε'-optimal). If instead $|\mathcal{L}_{\mathtt{NR}}| = \infty$, as in Lemma 4.2, $\|z_{*,\ell}\|^2 \to 0$; hence, $0 \in \partial_{\alpha_{*,\infty}} f(\bar{x})$ by (2.18). For the bound on $\alpha_{*,\infty}$ just note that $\alpha_{*,\ell} \geq \alpha_{*,\infty}$ for ℓ is large enough and use (4.3). □

Hence, an infinite sequence of NR and NS proves that the last stability center $\bar{x}$—that none of the two kind of step changes—is "as optimal as it gets," not only approximately so. As the proof shows, the obtained accuracy can actually be higher than ε': by chance the last SS may have even found a bona fide optimal solution, that a final sequence of NS/NR will then certify as such ($\alpha_{*,\ell} \searrow 0$). To streamline the final discussion, it is useful to remark that the algorithm "aims" at accuracy ε', which is arguably (approximately) the best one that can be achieved [9, Observation 2.7]; hence, setting $\delta_2 < \varepsilon'$ would not make sense. Reversing this logic, if $\delta_2 > \varepsilon' = \bar{\varepsilon}/(m_2 - m_1)$, one can use $(m_2 - m_1)\delta_2$ in place of $\bar{\varepsilon}$ in Step 2.1. This corresponds to the fact, that, intuitively, if a δ_2-optimal solution is sought for, then it might not make sense to accept a decrease in function value (much) smaller than δ_2. With this expedient the following holds.

THEOREM 4.5. *If all oracles are informative boundedly uncooperative, Step* 2.1 *is in force, and Assumptions* 2, 3, 5*, and* 6 *hold, then either* $\hat{f}_\infty = -\infty$*, or the algorithm (finitely, if* $\delta_1 > 0$*) determines a(n approximately)* δ_2*-optimal solution.*

Proof. Step 2.1 and Assumption 5 ensure that the algorithm is well-defined: either an NR is done, or Assumption 1 holds. If $\hat{f}_\infty = -\infty$, there is nothing to prove; otherwise by Lemma 4.3 the number of SS is finite. After the last one, if $\delta_1 = 0$, the algorithm can still run forever, but Theorem 4.4 proves that $\bar{x}$ is δ_2-optimal. By the same arguments, if $\delta_1 > 0$, then the algorithm finitely stops, either in Step 2 or in Step 2.1, in both cases proving that $\bar{x}$ is approximately δ_2-optimal (cf. Lemma 4.1). ∎

We remark that implementing Step 2.1 with $\delta_2 > 0$ in place of $\bar{\varepsilon}$ is possible (and, in principle, makes sense) even with a cooperative oracle. This kind of handling of t_ℓ (termed, e.g., "long-term t-strategy" in [8]) has been shown to be actually useful to improve the practical behavior in some applications.

4.2. Uninformative faithful boundedly uncooperative oracles. A second kind of oracle closely mirrors the previous one, but with a significant difference: it satisfies (4.1) with an *unknown* $\bar{\varepsilon}^k < \infty$. This is, e.g., the case of solution methods for (1.3^k_x) that can provide upper and lower bounds but cannot (or are not willing to) guarantee that these will be arbitrarily close, such as exact methods with a tight bound on the computational resources. While most of the properties discussed in the previous section are still true for this *uninformative* (but still faithful) oracle, Step 2.1 can no longer be used: whether or not $\varepsilon^k \geq \bar{\varepsilon}^k$ can only be verified *ex-post* from the output of $\mathcal{O}^k$. This is why an "escape clause" is now needed in the inner loop:

Step 3′ (Termination) `if` neither (3.5) nor (3.6) hold, `then if` some oracle $\mathcal{O}^k$ still has to be called with $\varepsilon^k = \beta^k \varepsilon$, `then go to` Step 1;

Conversely, there is no way in which Assumption 5 could be guaranteed to hold a priori, and hence no need for it. Correspondingly, the NR step in the main loop needs to move after the oracle interaction:

Step 4.1 (Noise reduction) `If` neither (3.5) nor (3.6) hold, `then if` $\|z_{*,\ell}\|^2 \leq \delta_1$, `then stop else` change t_ℓ appropriately; $\bar{x}_{\ell+1} \hookleftarrow \bar{x}_\ell$;

In the rest of this section we will assume that Step 3′ and Step 4.1 are in effect. As in the previous case, stopping in the NR step is justified by the fact that $\bar{x}_\ell$ is then already "as optimal as it gets."

Lemma 4.6. *If the algorithm stops in Step* 4.1*, then* $\bar{x}_\ell$ *is* approximately $\bar{\varepsilon}/(m_2 - m_1)$*-optimal.*

Proof. Just use the fact that for each $\ell \in \mathcal{L}_{\mathtt{NR}}$, both (3.5) and (3.6) fail: subtracting their opposites and using (4.1) yields $\infty > \bar{\varepsilon} \geq \bar{f}_{\ell+1} - \underline{f}_{\ell+1} \geq \hat{f}_{\ell+1} - \check{f}_{\ell+1} \geq \overline{\mathtt{tar}}_\ell - \underline{\mathtt{tar}}_\ell = \varepsilon_\ell$, and then one can proceed exactly as in Lemma 4.1. ∎

We remark that also this NR, like Step 2.1 in paragraph 4.1 is most definitely different from that originally proposed in [27], which will be described in section 4.3. Despite being triggered differently, the two NR work in the same way; for instance, Lemma 4.2 is clearly still valid. The subtle difference is that working ex-post implies a somewhat "more optimistic" stance: although the oracles may not bound themselves to deliver any required accuracy, they may actually happen to be able to do so, ultimately yielding an exact solution. In other words, as previously remarked, just changing t_ℓ gives the oracles another chance, and this may in itself be enough to "unstuck" the algorithm. In terms of convergence analysis, this corresponds to the fact that, unlike with Lemma 4.3, there is now nothing preventing a priori $v_{*,\ell} \to 0$, and therefore $|\mathcal{L}_{\mathtt{SS}}| = \infty$ may happen. In this case Theorem 3.1 still applies, as indeed it does not depend on what happens between two consecutive SS, be them NR or NS. Hence, Assumption 6 can be relaxed to hold only when finitely many SS are performed.

Assumption 7. *If* $|\mathcal{L}_{\mathtt{SS}}| < \infty$ *and* $|\mathcal{L}_{\mathtt{NR}}| = \infty$*, then* $\liminf_{\ell \ni \mathcal{L}_{\mathtt{NR}} \to \infty} t_\ell = \infty$.

Usefully, the simple implementation scheme of section 4.1, where $t_{\max}$ is reset (potentially) at every SS, also provides the relaxed Assumption 7. This means that, provided that $|\mathcal{L}_{\mathtt{SS}}| < \infty$, Lemma 4.2 and Theorem 4.4 still work. All in all, only minor modifications are required to the global result.

THEOREM 4.7. *Assume that all oracles are uninformative faithful boundedly uncooperative, Step* 4.1 *is in force, and Assumptions* 2, 3, *and* 7 *hold. Barring the obvious case* $\hat{f}_\infty = -\infty$*, if* $\delta_1 > 0$ *and* $\delta_2 > 0$*, then the algorithm finitely terminates with an* approximately $\max\{\varepsilon', \delta_2\}$*-optimal solution. If* $\delta_1 = 0$*, then* $\{\, \bar{x}_\ell \,\}$ *converges (possibly, finitely) to a* δ_2*-optimal solution.*

Proof. Clearly, whenever an SS or an NS is performed Assumption 1 is satisfied. If $\delta_1 > 0$ and $\delta_2 > 0$, then from Theorem 3.1 $|\mathcal{L}_{\mathtt{SS}}| < \infty$. Similarly, Lemma 4.2 implies that $|\mathcal{L}_{\mathtt{NR}}| < \infty$. Finally, Theorem 3.3 implies that $|\mathcal{L}| < \infty$ as well, for otherwise there would be a sequence of infinite consecutive NS. Hence the algorithm finitely terminates: if it stops in Step 2, then $\bar{x}_\ell$ is (approximately) δ_2-optimal; otherwise it stops in Step 4.1 and therefore $\bar{x}_\ell$ is (approximately) ε'-optimal by Lemma 4.6.

When $\delta_1 = 0$, the oracle may still happen to behave "cooperatively enough": if $|\mathcal{L}_{\mathtt{SS}}| = \infty$ or an infinite sequence of consecutive NS is performed, then the proof proceeds as in Theorem 3.4, with $\{\, \bar{x}_\ell \,\}$ converging (possibly, finitely) to a solution that is at least δ_2-optimal. Otherwise, the algorithm running forever implies $|\mathcal{L}_{\mathtt{NR}}| = \infty$, and then Theorem 4.4 applies. ∎

4.3. Uninformative cheating boundedly uncooperative oracles.

The last relevant case, in terms of solution methods for (1.3^k_x), is the one of heuristics without any known bound, even a posteriori, on the accuracy. Hence, while the oracle actually satisfies (4.1), it does so not only with an unknown $\bar{\varepsilon}^k < \infty$ (that is, the oracle is still boundedly uncooperative), but also with an *unknown* $\bar{f}^k$. We remark that this is the "minimalistic" stance typically taken in the literature, from [27] to [12]. In our context, this has the difficulty that the only *available correct* upper estimate, ∞, cannot ever satisfy (2.2)(i) whenever $\varepsilon^k < \infty$; in other words, *no significant upper model* $f^k_{\mathcal{P}}$ can be defined if $\bar{f}^k = \infty$. A possible resort is *cheating*, i.e., reporting incorrect data. An obvious way of doing that is systematically reporting $\bar{f}^k = \underline{f}^k$, which has the advantage that, at least purportedly, (2.2)(i) is satisfied for every ε^k. As we shall see this is actually, in some sense, the best possible approach, and we define *uniformly cheating* an oracle doing just that. The consequence of (4.1) is that $f^k \leq \check{f}^k_{\mathcal{B}} + \bar{\varepsilon}^k$: via (2.8), this gives

$$f^k(\bar{x}) - [\, \underline{f}^k_i + z^k_i(\bar{x} - x_i) \,] \leq \bar{f}^k + \bar{\varepsilon}^k - [\, \underline{f}^k_i + z^k_i(\bar{x} - x_i) \,] = \alpha^k_i + \bar{\varepsilon}^k, \tag{4.4}$$

$$z^k_i \in \partial_{(\alpha^k_i + \bar{\varepsilon}^k)} f^k(\bar{x}) \ \ , \ \ \alpha^k_i \geq -\bar{\varepsilon}^k \ \ , \ \ z_{*,\ell} \in \partial_{(\alpha_{*,\ell} + \bar{\varepsilon})} f(\bar{x}) \ \ , \ \ \alpha_{*,\ell} \geq -\bar{\varepsilon} \tag{4.5}$$

(with $\bar{\varepsilon} = \bar{\varepsilon}^{\mathcal{K}}$). Since the $\bar{\varepsilon}^k$ are not known, but are finite, the quality of the z^k_i cannot be directly assessed, but at least the error is bounded. Furthermore, the advantage of uniformly cheating oracles $\bar{f}^k = \underline{f}^k$ is that *they never trigger an NR Step* 4.1: Whatever the returned value, one of the two targets is surely met, and $\bar{f}^k - \underline{f}^k = 0 \leq \varepsilon^k$. Of course, inexactness has to crop up some other way; in particular, it may happen that $\check{f}^k_\ell(\bar{x}_\ell) > \hat{f}^k_\ell(\bar{x}_\ell)$, i.e., $\alpha_{*,\ell} < 0$. This may ultimately lead to $v_{*,\ell} = -t_\ell \left\| z_{*,\ell} \right\|^2 - \alpha_{*,\ell} > 0$, basically destroying all previous convergence arguments. The well documented remedy in the literature is to ensure that $\alpha_{*,\ell}$ is "not too negative":

$$\alpha_{*,\ell} \geq -m_3 t_\ell \left\| z_{*,\ell} \right\|^2 \tag{4.6}$$

with $m_3 \in (0,1)$, which immediately implies

$$v_{*,\ell} \leq (1-m_3)\big(-t_\ell \|z_{*,\ell}\|^2\big) \ \ (<0) \ . \tag{4.7}$$

Remarkably, this is similar but different from the condition always used in the literature, from the original [27] and up until the recent [12, (5.1)] [37], which is stricter in that it requires $m_3 \leq 1/2$, whereas we will argue that $m_3 \approx 1$ is preferable. It has been observed in [37] that m_3 influences the definition of predicted decrease; however, our analysis makes it apparent—due to the fact that, in our setting, (3.5) and (3.6) are not mutually exclusive—that the effect is actually on the NS condition. The fact that $\alpha_{*,\ell} < 0$ may happen implies that the NR now has to be performed at the end of Step 1, i.e., before the standard stopping condition which, being based on (2.18), is in principle no longer valid:

Step 1.1 (Noise reduction) `if` (4.6) fails, `then if` $\|z_{*,\ell}\|^2 \leq \delta_1$, `then stop else` { change t_ℓ according to Assumption 7; $\bar{x}_{\ell+1} \hookleftarrow \bar{x}_\ell$; $\mathcal{B}_{\ell+1} = \mathcal{B}_\ell$; $\mathcal{P}_{\ell+1} = \mathcal{P}_\ell$; $\ell \hookleftarrow \ell+1$; `go to` Step 1 };

Hence, this is again an "ex ante" NR, similarly to Step 2.1 in section 4.1 and quite unlike the "ex post" Step 4.1 in section 4.2. All three forms are, however, significantly different from each other. Step 2.1 is "proactive": it entirely avoids calling the oracles with data which may trigger problems due to the lack of accuracy. Step 4.1 rather reacts to the fact that the oracles, in the *current* iterate, have already produced correct but not accurate enough information. Step 1.1 instead reacts to the fact that the oracles, in some *previous* iterate, have produced incorrect information which makes $x_{\ell+1}$ not a significant point to even call the oracle upon. Yet, the convergence arguments are similar to those of the previous sections, accounting for the technical differences, starting from the fact that the stopping condition in the NR step is sound (cf. Lemmas 4.1 and 4.6).

LEMMA 4.8. *If the algorithm stops in Step* 1.1*, then* $\bar{x}_\ell$ *is* approximately *$\bar{\varepsilon}$-optimal.*

Proof. First note that (4.5) gives $z_{*,\ell} \in \partial_{\alpha_{*,\ell}+\bar{\varepsilon}} f(\bar{x}_\ell)$, which together with $\alpha_{*,\ell} + \bar{\varepsilon} \geq 0$ and $\|z_{*,\ell}\|^2 \leq \delta_1$ (as a result of stopping in Step 1.1) implies $(\alpha_{*,\ell}+\bar{\varepsilon})$-optimality of $\bar{x}_\ell$. Since $\ell \in \mathcal{L}_{\mathtt{NR}}$ implies $\alpha_{*,\ell} < 0$, $\bar{x}_\ell$ is therefore a fortiori $\bar{\varepsilon}$-optimal. □

However, the rest of the analysis requires specific development, since many arguments in the previous sections use (2.18) and therefore require $\alpha_{*,\ell} \geq 0$. This is in particular true for Theorem 3.1, which is based on the results of [7]. Its replacement is the following.

PROPOSITION 4.9. *Under the assumptions of Theorem* 3.1*, with $\hat{f}_\infty$ defined as in Theorem* 4.4*, it holds that* $\hat{f}_\infty = \limsup_{\mathcal{L}_{\mathtt{SS}} \ni \ell \to \infty} \check{f}_\ell(\bar{x}_\ell) \leq \nu(1.1) + \bar{\varepsilon}$.

Proof. When $\bar{f}_\infty > -\infty$, (3.4), (3.5) together with (4.7) give $v_{*,\ell} \to 0$. Since $\ell \in \mathcal{L}_{\mathtt{SS}}$ one has that (4.7) holds, which gives (barring the case $\hat{f}_\infty = -\infty$) that $\|z_{*,\ell}\| \to 0$, where we recall $\sum_{\ell \in \mathcal{L}_{\mathtt{SS}}} t_\ell = \infty$, by Assumption 2(ii) and the argumentation following (3.8). Consequently, (4.6) gives $\alpha_{*,\ell} \to 0$, even if $\alpha_{*,\ell} < 0$. This proves the statement about the limits being equal. Hence, if the sequence $\{\bar{x}_\ell\}_{\ell \in \mathcal{L}_{\mathtt{SS}}}$ has a cluster point $\bar{x}_\infty$, (4.5) gives $0 \in \partial_{\bar{\varepsilon}} f(\bar{x}_\infty)$, i.e., $\bar{x}_\infty$ is $\bar{\varepsilon}$-optimal. The proof can be generalized without requiring $\{\bar{x}_\ell\} \to \bar{x}_\infty$ by using, e.g., [12, Proposition 6.1], albeit at the cost of requiring t_ℓ to be bounded away from zero (cf. the discussion of [12, Theorem 6.2]). □

The equivalent of Lemma 4.2/Theorem 4.4 also holds.

PROPOSITION 4.10. *If all oracles $\mathcal{O}^k$ are uniformly cheating, Assumption* 7 *holds,* $|\mathcal{L}_{\mathtt{SS}}| < \infty$ *and* $|\mathcal{L}_{\mathtt{NR}}| = \infty$*, and then* $\limsup_{\ell \ni \mathcal{L}_{\mathtt{NR}} \to \infty} \|z_{*,\ell}\|^2 = 0$.

Proof. Use (4.5) and (2.15) to obtain $v_{*,\ell} \leq -t_\ell \|z_{*,\ell}\|^2 + \bar\varepsilon$, and then since (4.6) fails, the opposite of (4.7) holds and allows us to derive that $\bar\varepsilon > m_3 t_\ell \|z_{*,\ell}\|^2$: $\|z_{*,\ell}\|^2 \geq \delta > 0$ contradicts $t_\ell \to \infty$. □

PROPOSITION 4.11. *Under the assumptions of Proposition* 4.10*, $\bar x$ is $\bar\varepsilon$-optimal.*

Proof. Proposition 4.10 gives $\|z_{*,\ell}\| \to 0$: hence $0 \in \partial_{(\alpha_{*,\infty}+\bar\varepsilon)} f(\bar x)$. Therefore $\bar x$ is $(\alpha_{*,\infty}+\bar\varepsilon)$-optimal, and since $\alpha_{*,\ell} < 0$ for all $\ell \in \mathcal{L}_{\mathtt{NR}}$, $\alpha_{*,\infty} \leq 0$. □

Remark 4.12. The above proof relies on $\mathcal{O}$ to be uniformly cheating: $\alpha_{*,\ell} < 0$ is true because only Step 1.1 triggers an NR. If $\mathcal{O}$ is not uniformly cheating, Step 4.1 may happen, so $\alpha_{*,\ell} < 0$ might not be true. Using (4.3), which holds when $t_\ell \to \infty$, would give $0 \in \partial_{2\bar\varepsilon} f(\bar x)$, i.e., the error would be twice that of the oracle. This can actually happen, as illustrated by the following example. Let $f(x) = |x|$ and $\bar x_1 = x_1 = -2$. A (uniformly) cheating oracle with $\bar\varepsilon = 1$ may return $\underline{f}_1 = \bar f_1 = 1 < f(x_1) = 2$ and $z_1 = -1$. If $t = 1$, $x_2 = -1$, $v_{*,1} = -1$, and $\check f_1(\bar x_2) = 0$. A nonuniform oracle may now work "in reverse" on x_2, returning $\underline{f}_2 = 0$, $z_2 = 0$, and $\bar f_2 = 1 = f(x_2)$. Neither an SS ($\bar f_1 = \bar f_2$) nor an NS ($\underline{f}_2 = \check f_{\mathcal{B}_1}(\bar x_2)$) can be performed, so an NR Step 4.1 is triggered. However, with any $t_\ell > 1$, $x_2 = -1$ is a minimum of $\check f_2$, and hence $x_\ell = x_2$ for all ℓ may happen: the algorithm might never be able to improve $\bar x_1$, which has error $f(\bar x_1) - \nu(1.1) = 2 - 0 = 2 = 2\bar\varepsilon$. Note that if the minimum of f had been < 0, the oracle could have not reported $z_2 = 0$ together with $\underline{f}_2 = 0 = \check f_2(\bar x_2)$: $z_2 < 0$ would have been required, and therefore increasing t_ℓ would have ultimately triggered an SS. Therefore, once again $|\mathcal{L}_{\mathtt{NR}}| = \infty$ asymptotically proves $\bar\varepsilon$-optimality of $\bar x$. The case of the infinite tail of consecutive NS is slightly more complex, because a modification to the algorithm is needed: one has to *replace $\Delta_{*,\ell}$ with $-v_{*,\ell}$* in the definition of the lower targets $\underline{\mathtt{tar}}_\ell/\underline{\mathtt{tar}}^k$, and therefore of the accuracy ε_ℓ:

$$\underline{\mathtt{tar}}_\ell := \check f_\ell(x_{\ell+1}) - m_1 v_{*,\ell} \quad , \tag{4.8}$$

$$\varepsilon_\ell := \overline{\mathtt{tar}}_\ell - \underline{\mathtt{tar}}_\ell = (m_2 - m_1)(-v_{*,\ell}) \quad , \tag{4.9}$$

$$\underline{\mathtt{tar}}^k := \check f_+^k - m_1 \beta^k v_* \quad . \tag{4.10}$$

The reason is that while (4.6) ensures that $-v_{*,\ell} > 0$ (cf. (4.7)), $\Delta_{*,\ell} < 0$ still can happen. This would lead to $\underline{\mathtt{tar}}_\ell < \check f_\ell(x_{\ell+1})$, and to (3.6) being always satisfied, which would break all the development. Since $-v_{*,\ell} \geq \Delta_{*,\ell}$, (4.10) is harder to satisfy than (3.6), which is why—in the tradition of the standard analysis of bundle methods—we prefer to keep the weaker condition if possible. However, cheating oracles require (4.10), at least with "large" m_3; restricting to $m_3 < 1/2$ as in [12, equation (5.1)] would allow using (3.6) instead.

PROPOSITION 4.13. *Using* (4.8)–(4.10)*, and under the Assumptions of Theorem* 3.3 *plus $|\mathcal{L}_{\mathtt{NR}}| < \infty$, $\|z_{*,\ell}\| \to 0$ and $\alpha_{*,\ell} \to 0$: therefore, $\bar x$ is $\bar\varepsilon$-optimal.*

Proof. The crucial point is showing that $\limsup_{\ell\to\infty} v_{*,\ell} = 0$, so assume that $v_{*,\ell} \leq -\varepsilon < 0$: then (4.10) would imply that $\Delta f^{\mathcal{Z}} \geq -m_1 v_* \geq m_1 \varepsilon > 0$, and hence, by Lemma 3.2, $\nu(2.14_\ell) \to -\infty$ as $\ell \to \infty$, contradicting $\nu(2.14) \geq -\bar\varepsilon > -\infty$ (the latter a consequence of (4.5)). As in the SS case, $v_{*,\ell} \to 0$ gives $\|z_{*,\ell}\| \to 0$ via (4.7), and then (4.6) implies $\alpha_{*,\ell} \to 0$: hence, $0 \in \partial_{\bar\varepsilon} f(\bar x)$, i.e., $\bar x$ is $\bar\varepsilon$-optimal. □

The final result is then standard.

THEOREM 4.14. *Assume that all oracles are uninformative uniformly cheating boundedly uncooperative, Step* 1.1 *is in force, and Assumptions* 2, 3, *and* 7 *hold. Barring the obvious case $\hat{f}_\infty = -\infty$, if $\delta_1 > 0$ and $\delta_2 > 0$, then the algorithm finitely terminates with an* approximately $\max\{\bar{\varepsilon}, \delta_2\}$*-optimal solution. If $\delta_1 = 0$, then $\{\, \bar{x}_\ell \,\}$ converges (possibly, finitely) to an δ_2-optimal solution.*

Proof. Because all oracles are uniformly cheating, Assumption 1 clearly is fulfilled. Combine Theorem 4.9 and Propositions 4.11 and 4.13 as usual to cover the cases $|\mathcal{L}_{\mathtt{SS}}| = \infty$, $|\mathcal{L}_{\mathtt{NR}}| = \infty$, and the final tail of infinitely many NS, respectively, for $\delta_1 = 0$; finite convergence when $\delta_1 > 0$ is as usual an immediate corollary. □

While the above derivation closely mirrors that of the previous sections, all the results only involve the a priori (unknown) error $\bar{\varepsilon}$: no a posteriori bounds are available. It could not be otherwise, since no valid explicit upper estimates for the f-values are available. The final remark is that there is no need to analyze the concept of *informative cheating* oracle: if one would *know* $\bar{\varepsilon}^k$, then $\bar{f}^k = \underline{f}^k + \bar{\varepsilon}^k$ would be a valid upper bound, which would bring us fully back to section 4.1. This is actually the same as the controllable lower oracle of [12].

4.4. Arbitrary mixture of oracles. The previous sections may have conveyed the idea that faithful and cheating oracles cannot be used together: fortunately, this is not true. For simplicity we can disregard the difference between informative and uninformative faithful oracles, which is very easy to do: just disregard the available information about $\bar{\varepsilon}^k$. Also, note that cooperative oracles can also considered faithful with (known) $\bar{\varepsilon}^k = 0$. Hence, we can assume $\mathcal{K} = \mathcal{F} \cup \mathcal{C}$ ($\mathcal{F} \cap \mathcal{C} = \emptyset$) with $\mathcal{F}$ the components having faithful oracles, and $\mathcal{C}$ these having (uniformly) cheating ones. From (4.4) we have $z_{*,\ell} \in \partial_{(\alpha_{*,\ell} + \bar{\varepsilon}^{\mathcal{C}})} f(\bar{x})$ and $\nu(2.14) \geq \alpha_{*,\ell} \geq -\bar{\varepsilon}^{\mathcal{C}}$, whereas from the fact that all the cheating oracles are uniformly so we have $\bar{f}_{\ell+1} - \underline{f}_{\ell+1} \leq \bar{\varepsilon}^{\mathcal{F}}$. We of course now need to have both NR steps 1.1 and 4.1 (while we ignore step 2.1 for the time being): hence, $\mathcal{L}_{\mathtt{NR}} = \mathcal{L}^{\mathcal{F}}_{\mathtt{NR}} \cup \mathcal{L}^{\mathcal{C}}_{\mathtt{NR}}$, indicating respectively the NR of Step 4.1 ("for faithful oracles") and of Step 1.1 ("for cheating ones"); also, "NR" in Assumption 6 is now intended to mean "either kind of NR." Hence we can copy Lemma 4.2/Proposition 4.10.

PROPOSITION 4.15. *Under Assumption* 7, *if $|\mathcal{L}_{\mathtt{SS}}| < \infty$ and $|\mathcal{L}_{\mathtt{NR}}| = \infty$, then* $\limsup_{\ell \ni \mathcal{L}_{\mathtt{NR}} \to \infty} \|z_{*,\ell}\|^2 = 0$.

Proof. If $\ell \in \mathcal{L}^{\mathcal{F}}_{\mathtt{NR}}$, one can use $\bar{\varepsilon}^{\mathcal{F}} \geq \bar{f}_{\ell+1} - \underline{f}_{\ell+1}$ to get $\bar{\varepsilon}^{\mathcal{C}} > (m_2 - m_1) t_\ell \|z_{*,\ell}\|^2$ as in Lemma 4.2, while if $\ell \in \mathcal{L}^{\mathcal{C}}_{\mathtt{NR}}$, one can use $\alpha_{*,\ell} \geq -\bar{\varepsilon}^{\mathcal{C}}$ to get $\bar{\varepsilon}^{\mathcal{C}} > m_3 t_\ell \|z_{*,\ell}\|^2$ as in Proposition 4.10. Hence, in both cases, $\|z_{*,\ell}\|^2 \geq \delta > 0$ contradicts $t_\ell \to \infty$. □

Taking for simplicity $m_3 = m_2 - m_1$ and $\delta_2 = \bar{\varepsilon}(1 - m_3)/m_3$, we can now prove the following.

THEOREM 4.16 (convergence for arbitrary mixture of oracles). *Under Assumptions* 2–7 *and given any $\mathcal{F}$ and $\mathcal{C}$, the algorithm with both Steps* 1.1 *and* 4.1 *and using* (4.8)–(4.10) *(asymptotically) finds a(n approximated) $(\bar{\varepsilon}/m_3)$-optimal solution.*

Proof. If the algorithm finitely stops, this happens either in Step 1.1, in Step 2, or in Step 3. In all cases, $\bar{x}$ is approximately $(\alpha_{*,\ell} + \bar{\varepsilon}^{\mathcal{C}})$-optimal, in the sense that $\|z_{*,\ell}\| \leq \delta_1$. In the first case $\alpha_{*,\ell} < 0$ because $\ell \in \mathcal{L}^{\mathcal{C}}_{\mathtt{NR}}$; in the second, $\alpha_{*,\ell} \leq \delta_2$; in the third, $\alpha_{*,\ell} \leq \bar{\varepsilon}^{\mathcal{F}}$ because $\ell \in \mathcal{L}^{\mathcal{F}}_{\mathtt{NR}}$ (cf. (4.3)). Hence, $\bar{x}_\ell$ is always approximately $(\bar{\varepsilon}/m_3)$-optimal.

If $|\mathcal{L}_{\mathtt{SS}}| = \infty$, reasoning as in Proposition 4.9 proves that $\bar{f}_\infty \leq \nu(1.1) + \bar{\varepsilon}^{\mathcal{C}}$. Hence, we can assume the last SS to happen, after which only NR (of either type) or NS is performed: as in Proposition 4.15 $\bar{x}$ is $(\alpha_{*,\infty} + \bar{\varepsilon}^{\mathcal{C}})$-optimal. Now, if $\left|\mathcal{L}^{\mathcal{C}}_{\mathtt{NR}}\right| = \infty$, then $\alpha_{*,\infty} \leq 0$; otherwise $\alpha_{*,\infty} \leq \bar{\varepsilon}^{\mathcal{F}}/m_3$ as in Theorem 4.4; all in all, $\alpha_{*,\infty} + \bar{\varepsilon}^{\mathcal{C}} \leq \bar{\varepsilon}/m_3$. Finally, for the infinite tail of consecutive NS we can reason as in Proposition 4.13 to obtain that $\bar{x}$ is $\bar{\varepsilon}^{\mathcal{C}}$-optimal. □

If the cheating oracles are not uniformly so, we can no longer use the fact that $\bar{f}_{\ell+1} - \underline{f}_{\ell+1} \leq \bar{\varepsilon}^{\mathcal{F}}$; the bound is $\bar{f}_{\ell+1} - \underline{f}_{\ell+1} \leq \bar{\varepsilon}$. This bumps the worst-case error to $\bar{\varepsilon} + \bar{\varepsilon}^{\mathcal{F}} = \bar{\varepsilon}^{\mathcal{F}} + 2\bar{\varepsilon}^{\mathcal{C}}$: nonuniformly cheating oracles do "double damage." This is not entirely surprising, as they actually have "twice the range" of both faithful and uniformly cheating ones: they can report $\bar{f}^k = f^k(x) + \bar{\varepsilon}^k$ (and, then, $\underline{f}^k = f^k(x)$) as well as $\underline{f}^k = f^k(x) - \bar{\varepsilon}^k$ (and, then, $\bar{f}^k = f^k(x)$), and hence the range between the worst-case upper and lower estimate is indeed $2\bar{\varepsilon}^k$. An "adversarial" oracle can use it, as in Remark 4.12, to enforce an error of that magnitude.

We end by remarking that an even more convoluted approach would entail using Step 2.1 with lower estimates of $\bar{\varepsilon}$. These could be easily obtained by taking the maximum of $-\alpha_{*,\ell}$ when Step 1.1 is executed, together with the maximum of ε_ℓ when Step 4.1 is executed. At each time this is a lower estimate of the true $\bar{\varepsilon}$, which can be used to try to avoid unduly "short" steps; of course with no guarantee of success, but with Step 3′ in the inner loop and Step 4.1 readily available as fallbacks. The approach would clearly work, as the analysis of this section shows, and it may also have merits computationally. Whether or not this is the case can only be seen experimentally, which is left for future research.

5. Conclusions.

We have analyzed a novel class of bundle methods for minimizing a sum-structured convex nondifferentiable objective. Our starting point is to change the oracle definition so as to make apparent a feature that oracles often have in applications, but that has so far not been exploited: the fact that, besides lower estimates on the function value (and valid lower linearizations of the function epigraph), the oracles also provide *upper estimates.* The availability of this information has several notable impacts on the method, not only in the incremental case but also in the general inexact one (say, with only one oracle):

- It allows one to provide both *upper and lower targets* to the oracles which, together with the required accuracy, can allow them to stop their computation just as soon as information that is "accurate enough" to allow the algorithm to proceed is obtained, possibly diminishing their cost.
- It provides explicit and reliable a posteriori estimates of the quality of the obtained solutions.

Fully exploiting this in the incremental case requires a further Lipschitz continuity assumption—even more, that the Lipschitz constant is actually *known*—which, however, is often satisfied in many important applications. This, in turn

- allows one to define *upper models* of the functions, which can be used to derive upper estimates of function values even at points where some (or even all) of the oracles have not been called;
- allows one to skip oracle calls entirely for some of the component functions, not only at NS as in previous proposals in the literature, but also at SS.

The method works with oracles that cannot attain arbitrary precision, both if their maximum error is known or unknown, as well as with those that do not provide reliable upper estimates, clearly losing some of the properties in the process. We analyze basically all the (three) possible different forms of the (lower) oracle, as well as their combinations. Although the results are similar, the three different oracles require three significantly different forms of NR steps, as well as specific analysis. We show that for all kinds of oracles with nonzero (but bounded) maximum accuracy, one gets a solution with (about) the same error, which is the best that can be expected [9, Observation 2.7] and consistent with other methods in the literature; however, for oracles that do not provide reliable upper estimates, one loses the a-posteriori estimates of the quality of the obtained solutions. It is worth remarking that, in fact, reliable upper estimates provide something more: the fact that $\bar{f}_\ell$ is always a *reliable upper estimate on* $\nu(1.1)$, and *therefore on* $\nu(1.3)$. Computing tight upper estimates on $\nu(1.3)$ is very often the reason for solving (1.1) in the first place. For instance, in practical, large-scale industrial applications rarely problems are solved to optimality, and the decision maker can benefit from knowing whether or not it may be worthwhile to invest more to get better solutions. In general, if (1.1) is solved as a step of a more complex (e.g., implicit enumerative) approach to (1.3), valid upper estimates are often crucial. All this provides a compelling argument against "erasing" the provided upper estimate and setting $\bar{f}^k = \underline{f}^k$, as (implicitly) advocated in the literature so far, thus making oracles systematically (but, at least, uniformly) cheat, although this ultimately yields the same bound. Whether or not exploiting upper estimates works better computationally can only be determined by an in-depth experimental study, which is therefore the logical next step for this line of research.

Acknowledgment. We are indebted to the anonymous referees whose constructive comments have helped us to considerably improve the manuscript.

REFERENCES

[1] A. Astorino, A. Frangioni, A. Fuduli, and E. Gorgone, *A nonmonotone proximal bundle method with (potentially) continuous step decisions*, SIAM J. Optim., 23 (2013), pp. 1784–1809.

[2] A. Astorino, A. Frangioni, M. Gaudioso, and E. Gorgone, *Piecewise quadratic approximations in convex numerical optimization*, SIAM J. Optim., 21 (2011), pp. 1418–1438.

[3] A. Astorino, M. Gaudioso, and E. Gorgone, *A method for convex minimization based on translated first-order approximations*, Numer. Algorithms, (2017), pp. 1–16, https://doi.org/10.1007/s11075-017-0280-6.

[4] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, *Bundle methods in stochastic optimal power management: A disaggregate approach using preconditionners*, Comput. Optim. Appl., 20 (2001), pp. 227–244.

[5] O. Briant, C. Lemaréchal, P. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck, *Comparison of bundle and classical column generation*, Math. Program., 113 (2008), pp. 299–344.

[6] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos, *An augmented Lagrangian method for distributed optimization*, Math. Program., 152 (2015), pp. 405–434.

[7] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, Math. Program., 62 (1993), pp. 261–275.

[8] T. Crainic, A. Frangioni, and B. Gendron, *Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems*, Discrete Appl. Math., 112 (2001), pp. 73–99.

[9] G. d'Antonio and A. Frangioni, *Convergence analysis of deflected conditional approximate subgradient methods*, SIAM J. Optim., 20 (2009), pp. 357–386.

[10] W. de Oliveira and J. Eckstein, *A Bundle Method for Exploiting Additive Structure in Difficult Optimization Problems*, Optimization Online, 2015.

[11] W. de Oliveira and C. Sagastizábal, *Level bundle methods for oracles with on demand accuracy*, Optim. Methods Softw., 29 (2014), pp. 1180–1209.

[12] W. de Oliveira, C. Sagastizábal, and C. Lemaréchal, *Convex proximal bundle methods in depth: A unified analysis for inexact oracles*, Math. Program. Ser. B, 148 (2014), pp. 241–277.

[13] W. de Oliveira, C. Sagastizábal, and S. Scheimberg, *Inexact bundle methods for two-stage stochastic programming*, SIAM J. Optim., 21 (2011), pp. 517–544, https://doi.org/10.1137/100808289.

[14] L. Dubost, R. Gonzalez, and C. Lemaréchal, *A primal-proximal heuristic applied to French unitcommitment problem*, Math. Program., 104 (2005), pp. 129–151.

[15] G. Emiel and C. Sagastizábal, *Incremental like bundle methods with applications to energy planning*, Comput. Optim. Appl., 46 (2009), pp. 305–332.

[16] A. Frangioni, *Solving semidefinite quadratic problems within nonsmooth optimization algorithms*, Comput. Oper. Res., 21 (1996), pp. 1099–1118.

[17] A. Frangioni, *About Lagrangian methods in integer optimization*, Ann. Oper. Res., 139 (2005), pp. 163–193.

[18] A. Frangioni, C. Gentile, and F. Lacalandra, *Solving unit commitment problems with general ramp contraints*, Internat. J. Electr. Power Energy Syst., 30 (2008), pp. 316–326.

[19] A. Frangioni and E. Gorgone, *Generalized bundle methods for sum-functions with "easy" components: Applications to multicommodity network design*, Math. Program., 145 (2014), pp. 133–161.

[20] A. Frangioni, B. Gendron, and E. Gorgone, *On the computational efficiency of subgradient methods: A case study with Lagrangian bounds*, Math. Program. Comput., 9 (2017), pp. 573–604, https://doi.org/10.1007/s12532-017-0120-7.

[21] M. Gaudioso, G. Giallombardo, and G. Miglionico, *An incremental method for solving convex finite min-max problems*, Math. Oper. Res., 31 (2006), pp. 173–187.

[22] C. Helmberg, *Numerical experiments on dynamically choosing cutting models in bundle methods*, in Proceedings of the International Workshop on Optimization Challenges in the Evolution of Energy Networks to Smart Grids, 2016.

[23] C. Helmberg and S. Röhl, *A case study of joint online truck scheduling and inventory management for multiple warehouses*, Oper. Res., 55 (2007), pp. 733–752.

[24] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* I, 2nd ed., Grundlehren Math. Wiss. 305, Springer-Verlag, Berlin, 1996.

[25] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* II, 2nd ed., Grundlehren Math. Wiss. 306, Springer-Verlag, Berlin, 1996.

[26] K. Jones, I. Lustig, J. Farwolden, and W. Powell, *Multicommodity network flows: The impact of formulation on decomposition*, Math. Program., 62 (1993), pp. 95–117.

[27] K. Kiwiel, *A proximal bundle method with approximate subgradient linearizations*, SIAM J. Optim., 16 (2006), pp. 1007–1023.

[28] C. Lemaréchal, *Lagrangian relaxation*, in Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions, M. Jünger and D. Naddef, eds., Lecture Notes in Comput. Sci. 9, Springer-Verlag, Berlin, 2001, pp. 112–156.

[29] C. Lemaréchal and A. Renaud, *A geometric study of duality gaps, with applications*, Math. Program., 90 (2001), pp. 399–427.

[30] Y. Song and J. Luedtke, *An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse*, SIAM J. Optim., 25 (2015), pp. 1344–1367, https://doi.org/10.1137/140967337.

[31] M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra, *Large-scale unit commitment under uncertainty: A literature survey*, 4OR, 13 (2015), pp. 115–171, https://doi.org/10.1007/s10288-014-0279-y.

[32] W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal, *Probabilistic optimization via approximate p-efficient points and bundle methods*, Comput. Oper. Res., 77 (2017), pp. 177–193, https://doi.org/10.1016/j.cor.2016.08.002.

[33] W. van Ackooij and W. de Oliveira, *Level bundle methods for constrained convex optimization with various oracles*, Comput. Optim. Appl., 57 (2014), pp. 555–597.

[34] W. van Ackooij, W. de Oliveira, and Y. Song, *An adaptive partition-based level decomposition for solving two-stage stochastic programs with fixed recourse*, Informs J. Comput., 130 (2017), pp. 57–70, https://doi.org/10.1287/ijoc.2017.0765.

[35] W. van Ackooij, A. Frangioni, and W. de Oliveira, *Inexact stabilized Benders' decomposition approaches: With application to chance-constrained problems with finite support*, Comput. Optim. Appl., 65 (2016), pp. 637–669, https://doi.org/10.1007/s10589-016-9851-z.

[36] W. van Ackooij and J. Malick, *Decomposition algorithm for large-scale two-stage unit-commitment*, Ann. Oper. Res., 238 (2016), pp. 587–613, https://doi.org/10.1007/s10479-015-2029-8.

[37] W. van Ackooij and C. Sagastizábal, *Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems*, SIAM J. Optim., 24 (2014), pp. 733–765.

[38] C. Wolf, C. I. Fábián, A. Koberstein, and L. Stuhl, *Applying oracles of on-demand accuracy in two-stage stochastic programming—A computational study*, European J. Oper. Res., 239 (2014), pp. 437–448.