

BUNDLE METHODS IN THE XXIst CENTURY: A BIRD'S-EYE VIEW

Welington de Oliveira¹ and Claudia Sagastizábal^{2*}

Received September 30, 2013 / Accepted December 12, 2013

ABSTRACT. Bundle methods are often the algorithms of choice for nonsmooth convex optimization, especially if accuracy in the solution and reliability are a concern. We review several algorithms based on the bundle methodology that have been developed recently and that, unlike their forerunner variants, have the ability to provide exact solutions even if most of the time the available information is inaccurate. We adopt an approach that is by no means exhaustive, but covers different proximal and level bundle methods dealing with inexact oracles, for both unconstrained and constrained problems.

Keywords: bundle methods, inexact oracles, nonsmooth convex optimization.

1 INTRODUCTION

Nonsmooth convex optimization (NCO) appears often in connection with real-life problems that are too difficult to solve directly and need to be decomposed. Typically, Lagrangian relaxation or Benders' decomposition approaches replace the hard problem by a sequence of simpler problems, corresponding, in general, to an iteration of some nonsmooth convex method.

As in Nonlinear Programming, NCO methods define iterates using *oracle* information, which in the nonsmooth setting comes in the form of a function value and a subgradient. When dealing with the hard problems mentioned above, usually the oracle information cannot be computed exactly (the procedure may be too time consuming, or just impossible, for example, if differential equations or continuous stochastic processes are involved). We refer to this situation as dealing with an *inexact oracle*; for many examples in the energy sector, see [49].

In 1975 Claude Lemaréchal and Philip Wolfe independently created bundle methods to minimize a convex function for which only one subgradient at a point is computable, [29] and [54]. The short "historical" story [39] discusses the different developments that resulted in the $\mathcal{V}\mathcal{U}$ -bundle variants [38], that are superlinearly convergent with respect to so-called serious steps. Such a

*Corresponding author.

¹Instituto Nacional de Matemática Pura e Aplicada – IMPA, 22460-320 Rio de Janeiro, Brazil. E-mail: wlo@impa.br

²Visiting researcher. E-mail: sagastiz@impa.br

convergence rate was once deemed unattainable for nonsmooth functions, thus explaining the “science fiction” wording in the title of [39].

Since their introduction, and for practically 25 years, the convergence theory of bundle methods could only handle *exact* oracles. It was only in 2001 that the work [17] opened the way to inexactness. It was followed by [53] and [27].

For a long time bundle algorithms have been considered as the methods of choice for NCO problems requiring reliable and accurate solutions. The added ability of handling inexact oracles made the new variants even more interesting for real-life applications, especially those involving stochastic programming models. Works such as [43, 44, 2] give a flavor of the many different ways oracle inexactness can be exploited within a bundle method – to eventually solve the hard problem in an *exact* way. As shown by the numerical experience in [43], for a large battery of two-stage stochastic linear programs it is possible to save up to 75% of the total computational time (while still finding an exact solution) just by letting the bundle method control how accurate the oracle information should be at consecutive iterates. The main idea is that there is no gain in “wasting” time by computing exact oracle information at the so-called null steps; we do not enter into more details here, but refer instead to [43].

In this work we focus on the main ideas behind several variants of bundle methods, rather than on technicalities or fine details. After laying down a minimal background in Section 2, we consider separately unconstrained and constrained methods in Section 3 and 4, respectively. To illustrate the importance of modeling when solving real-life optimization problems, Section 5 contains an example of hydroreservoir management that can be cast into two different formulations, calling for unconstrained or constrained methods. Section 6 concludes with some closing remarks and comments on future research.

2 ORACLE ASSUMPTIONS AND ALGORITHMIC PATTERN

As explained in [48], the versatility of bundle methods dealing with inexact oracles is best exploited when the *structure* of the NCO problem is taken into account. For this reason, we consider below two structured optimization problems, defined over a polyhedral set $\mathcal{X} \subset \Re^n$ and involving two convex functions, $f : \Re^n \rightarrow \Re$ and $h : \Re^n \rightarrow \Re$, possibly nonsmooth:

- unconstrained or linearly constrained problem

$$v_{\text{opt}} := \min_x f(x) + h(x) \quad \text{s.t. } x \in \mathcal{X}. \quad (1a)$$

- problem with nonlinear convex constraints

$$v_{\text{opt}} := \min_x f(x) \quad \text{s.t. } x \in \mathcal{X} \quad \text{and } h(x) \leq 0. \quad (1b)$$

The interest in considering separately the two functions f and h lies in their oracles. More precisely, the function f is *easy*: for each given $x^k \in \mathcal{X}$ the f -oracle provides

$$\begin{cases} \text{the exact value of the function} & f(x^k), \\ \text{and an exact subgradient} & g^k \in \partial f(x^k). \end{cases} \tag{2}$$

In contrast, the function h is *hard* to evaluate and, hence, only approximate values are available. Accordingly, at any given point $x^k \in \mathcal{X}$ the h -oracle outputs the estimates

$$\begin{cases} h_x^k = h(x^k) - \eta_h^k, \\ s^k \in \mathfrak{N}^n \text{ such that } h(\cdot) \geq h_x^k + \langle s^k, \cdot - x^k \rangle - \eta_+, \end{cases} \tag{3}$$

where the errors η_h^k and η_+ can be unknown. It follows from the oracle definition that $h(x^k) \geq h_x^k + \langle s^k, x^k - x^k \rangle - \eta_+ = h(x^k) - (\eta_h^k + \eta_+)$, showing that $\eta_h^k + \eta_+ \geq 0$. Hence, the approximate subgradient is an $(\eta_h^k + \eta_+)$ -subgradient of h at x^k :

$$\forall x \in \mathfrak{N}^n \quad h(x) \geq h(x^k) + \langle s^k, x - x^k \rangle - (\eta_h^k + \eta_+) \quad \text{with } \eta_h^k + \eta_+ \geq 0.$$

An exact oracle corresponds to $\eta_h^k = \eta_+ = 0$. When only $\eta_+ = 0$, the oracle is said to be of the *lower* type: the linearization generated with the oracle information stays below the function everywhere. In contrast, when $\eta_+ > 0$, the oracle is of the upper type, and the linearizations may cut off portions of the graph of h . Several examples of such oracles are given in [44]; we just mention here that Lagrangian relaxation is a typical source of oracles of the lower type.

Algorithms in the bundle family use oracles (2) and (3) to define linearizations at iterates x^j , for instance

$$\begin{cases} \bar{f}_j(x) & := f(x^j) + \langle g^j, x - x^j \rangle \\ \bar{h}_j(x) & := h_x^j + \langle s^j, x - x^j \rangle \\ \overline{(f+h)}_j(x) & := f(x^j) + h_x^j + \langle g^j + s^j, x - x^j \rangle. \end{cases} \tag{4}$$

At the k -th iteration, the bundle is defined by past oracle information accumulated in some index set $J_k \subset \{1, 2, \dots, k\}$. Putting together the corresponding linearizations gives cutting-plane models, which can be built individually for each function, or for their sum:

$$\begin{cases} \check{f}_k(x) & := \max_{j \in J_k} \bar{f}_j(x) & (\leq f(x)) \\ \check{h}_k(x) & := \max_{j \in J_k} \bar{h}_j(x) & (\leq h(x) + \eta_+) \\ \overline{(f+h)}_k(x) & := \max_{j \in J_k} \overline{(f+h)}_j(x) & (\leq f(x) + h(x) + \eta_+). \end{cases} \tag{5}$$

By definition, the models above are piecewise linear convex functions.

The modeling functions are used to replace f and h in (1) and define iterates $\{x^k\} \subset \mathcal{X}$ at which the oracles will be called. Such a replacement can be done in different ways, to be specified later on. For now, and to cover both problems in (1), we just write the resulting k -th subproblem as the

minimization of an abstract objective function O_k , depending on a parameter set pars , specific to each considered bundle variant. Accordingly, the next iterate is given by

$$x^{k+1} \in \arg \min O_k(x; \text{pars}) \quad \text{s.t. } x \in \mathcal{X}. \quad (6)$$

Typically, the bundle-variant-dependent set pars contains

- the convex model $\widetilde{(f+h)}_k$ of $f + h$ (or the sum of models \check{f}_k and \check{h}_k of f and h);
- a stability center \hat{x}^k , usually an element in the sequence $\{x^k\}$, together with a test to determine when the center can be updated (for example, to x^{k+1});
- an optimality measure to stop the iterations (such as upper and lower bounds for the optimal value of the problem);
- certain algorithmic parameters, to be updated at each iteration.

The main concern when specifying (6) is that the corresponding subproblem is easy, because it needs to be solved at each iteration of the bundle method. A simple illustration, not belonging to the bundle family but still fitting the abstract format above, is the cutting-plane method of [9] and [19], with subproblems fully characterized by

$$\text{pars} := \widetilde{(f+h)}_k \quad \text{and} \quad O_k(x; \text{pars}) := \widetilde{(f+h)}_k(x)$$

yielding a linear programming problem (LP) in (6):

$$x^{k+1} \in \arg \min \widetilde{(f+h)}_k(x) \quad \text{s.t. } x \in \mathcal{X}$$

(recall the set \mathcal{X} is defined by linear equality and inequality constraints.) Bundle methods can be seen as *stabilized* forms of the cutting-plane algorithm, well known for its instability and slow convergence. Different stabilization devices result in different bundle algorithms, such as the proximal variant [33] or the level one [30]. For these methods, the respective parameter sets pars involve more objects than just $\widetilde{(f+h)}_k$ and, as shown in Sections 3 and 4 below, give quadratic programming (QP) subproblems (6).

Once the parameter set and the modeling objective function in (6) are specified, the method needs to define updating rules for the elements in pars , including a stopping test. The main steps of all the algorithms are given below.

Algorithmic Pattern 1. *Oracles for f and h , a model O_k , and a rule for defining the parameter set pars are given.*

STEP 0. *For an initial iterate x^1 , call the oracles to obtain the information $(f(x^1), g^1)$ and (h_x^1, s^1) . Set $k = 1$ and initialize $J_1 = \{1\}$, $\hat{x}^1 = x^1$.*

STEP 1. *Solve (6) to obtain x^{k+1} .*

STEP 2. Perform a stopping test based on the optimality measure in *pars*.

STEP 3. If x^{k+1} was not considered close enough to optimal, call the oracles at x^{k+1} to obtain the information

$$(f(x^{k+1}), g^{k+1}) \text{ and } (h_x^{k+1}, s^{k+1}).$$

STEP 4. Depending on a test measuring progress towards the goal of solving (1), specified in *pars*, decide whether or not the stability center is to be replaced:

$$\text{set } \hat{x}^{k+1} \in \mathcal{X}.$$

STEP 5. Update *pars* by the given rule. Increase k by 1 and loop to Step 1. □

The above pattern is merely a sketch of an algorithm. Different models and different updating rules for the stability center define different methods. The stopping test performed in Step 2 depends, naturally, on the bundle method variant. In what follows we will make each Step in the Algorithmic Pattern 1 more precise, for different bundle method variants. In order to ease the presentation, we split our analysis into the two instances considered in (1), starting with the linearly constrained problem (1a).

3 UNCONSTRAINED OR LINEARLY CONSTRAINED SETTING

The algorithms considered in this section are suitable for problems with simple constraints (such as linear) or none at all, as in (1a). We start with the most well known variant, the proximal bundle method, [32, 25, 33, 15]; see also [37, 17, 53, 7, 38, 27, 24, 11, 45, 34, 44, 50].

3.1 Proximal Bundle Methods

This family defines subproblems based on the equivalence between minimizing a convex function and finding a fixed point of the proximal point operator, [40]:

$$\bar{x} \text{ solves (1a)} \iff \bar{x} \text{ solves } \min_{x \in \mathcal{X}} f(x) + h(x) + \frac{1}{2t} \|x - \bar{x}\|_M^2,$$

for some proximal stepsize $t > 0$ and a norm induced by a positive definite matrix M . To make the equivalence above implementable, methods in this variant replace \bar{x} by the current stability center and $f + h$ by some model, which can *aggregate* the linearizations and use $\overline{(f+h)}_k$, or *disaggregate* the information and use individual cutting-plane models for f and h :

$$\text{pars} = \begin{cases} \text{model} & \begin{cases} \overline{(f+h)}_k & \text{if aggregate variant} \\ \check{f}_k + \check{h}_k & \text{if disaggregate variant} \end{cases} \\ \text{stability center} & \hat{x}^k \in \{x^k\} \\ \text{stability function} & |\cdot|_M \\ \text{prox-stepsize} & t_k > 0. \end{cases}$$

Instead of the M -norm, the generalized bundle methods [15] use as stabilizing term a closed convex function satisfying certain properties.

The corresponding QP subproblems (6) are

- Aggregate model: $O_k(x; \text{pars}) = \widetilde{(f+h)}_k(x) + \frac{1}{2t_k} |x - \hat{x}^k|_M^2$.
- Disaggregate model: $O_k(x; \text{pars}) = \check{f}_k(x) + \check{h}_k(x) + \frac{1}{2t_k} |x - \hat{x}^k|_M^2$.

Since the objective function in both subproblems is strongly convex, the new iterate x^{k+1} defined by (6) is unique.

Depending on the structure of the problem to be solved, other modeling choices are possible. In [50], for instance, the hard function is the composition of a smooth mapping with a positively homogeneous convex function. The model in `pars` results from composing a cutting-plane model of the latter with a first Taylor expansion of the former (at the current stability center). Telecommunication networks often exhibit separable structure, that is exploited in [16] via a special Dantzig-Wolfe decomposition approach and in [31] by using a model that sums a cutting-plane model of the hard function and a second order expansion of the (smooth) easy term.

For simplicity in what follows we focus on a proximal bundle variant using the aggregate model $\widetilde{(f+h)}_k$ and $|\cdot|_M = |\cdot|$, the Euclidean norm.

Subproblem definition

We now provide all the necessary ingredients to make the Algorithmic Pattern 1 an implementable method, when the proximal subproblem uses the aggregate model, so that (6) becomes

$$x^{k+1} = \arg \min_{x \in X} \widetilde{(f+h)}_k(x) + \frac{1}{2t_k} |x - \hat{x}^k|^2, \tag{7}$$

and the unique solution to (7) is

$$x^{k+1} = \hat{x}^k - t_k(G^k + N^k) \text{ with } \begin{cases} G^k & := \sum_{j \in J_k} \alpha_j^k (g^j + s^j) \in \partial \widetilde{(f+h)}_k(x^{k+1}) \\ N^k & := -\frac{x^{k+1} - \hat{x}^k}{t_k} - G^k \in \partial i_X(x^{k+1}), \end{cases} \tag{8}$$

where i_X denotes the indicator function of the set X . As for the simplicial multiplier α^k in (8), it satisfies, for all $j \in J_k$:

$$\sum_{j \in J_k} \alpha_j^k = 1, \quad \alpha_j^k \geq 0, \quad \alpha_j^k [\widetilde{(f+h)}_k(x^{k+1}) - \overline{(f+h)}_j(x^{k+1})] = 0.$$

The complementarity relations above, in particular, allow to eliminate *a posteriori* inactive linearizations without losing relevant information but keeping the QP subproblems not too large.

The resulting *selection* mechanism takes in Step 5 of the Algorithmic Pattern 1 a bundle set satisfying

$$J_{k+1} \supset \{j \in J_k : \alpha_j^k > 0\} \cup \{k + 1\}$$

to define the next model $\overline{(f+h)}_{k+1}$ given in (5). An even more economical bundle, called *compressed*, can be built using the aggregate linearization defined in (11) below. As shown in [27] and [44], these strategies save storage without impairing convergence.

Stability Center and Optimality Certificate

The rule to update the stability center \hat{x}^k must assess the quality of the candidate x^{k+1} in terms of the goal of solving (1a). Therefore, if the iterate x^{k+1} provides enough decrease with respect to the “best” available value, $f(\hat{x}^k) + h_{\hat{x}}^k$, the stability center is changed to x^{k+1} . In the bundle jargon setting $\hat{x}^{k+1} = x^{k+1}$ is called making a “serious” step. This occurs when, for an Armijo-like parameter $m \in (0, 1)$,

$$f(x^{k+1}) + h_x^{k+1} \leq f(\hat{x}^k) + h_{\hat{x}}^k - m \left(f(\hat{x}^k) + h_{\hat{x}}^k - \overline{(f+h)}_k(x^{k+1}) \right). \tag{9}$$

The rightmost term above is a predicted improvement in the sense that it measures the distance between the best available value $f(\hat{x}^k) + h_{\hat{x}}^k$ and the best value predicted by the model, $\overline{(f+h)}_k(x^{k+1})$.

In this variant, the certificate of optimality depends on two terms, the predicted improvement employed in (9) and the direction, given by the sum of the gradient and the normal element in (8). The following object combines both terms

$$\phi_k := (f(\hat{x}^k) + h_{\hat{x}}^k) - \overline{(f+h)}_k(x^{k+1}) + \langle x^{k+1}, G^k + N^k \rangle, \tag{10}$$

and ensures convergence of the method whenever there exists a subsequence $\{(\phi_k, G^k + N^k)\}_{k \in K}$ that converges to $(\phi, 0)$, for some $\phi \leq 0$, [44, Theorem 3.2]. For this reason, an optimality measure for proximal bundle algorithms is to require both $\phi_k \leq 0$ and $|G^k + N^k|$ to be sufficiently small. When the serious-step sequence is bounded, this test boils down to the more traditional criterion, that checks if the approximate subgradient $G^k + N^k \in \partial_{\epsilon_k} \left((f+h) + i_{\mathcal{X}} \right) (\hat{x}^k)$ is small enough, where $\epsilon_k \geq 0$, which depends on the errors $\eta_h + \eta_+$, must also be sufficiently small.

Impact of inexactness

When the hard function h is exactly evaluated, or the oracle is of lower type, the definition of $G^k + N^k$ in (8) implies that the *aggregate linearization* stays always below the sum of functions:

$$\overline{(f+h)}_a(x) := \overline{(f+h)}_k(x^{k+1}) + \langle G^k + N^k, x - x^{k+1} \rangle \leq f(x) + h(x) \text{ for all } x \in \mathcal{X}. \tag{11}$$

In particular, the predicted improvement employed in (9) is always nonnegative for exact oracles and so is the difference $f(\hat{x}^k) + h_{\hat{x}}^k - \overline{(f+h)}_a(\hat{x}^k)$. In contrast, for lower and upper oracles the difference satisfies only the inequality

$$(f(\hat{x}^k) + h_{\hat{x}}^k) - \overline{(f+h)}_a(\hat{x}^k) \geq -(\eta_h^k + \eta_+).$$

Such difference can be negative, entailing difficulties in the convergence analysis. In order to overcome this drawback, whenever this difference is too negative the prox-parameter can be increased in Step 2 of the Algorithmic Pattern 1, as in [27]. The following related rule, based on a relative criterion, was considered in [3] and [44]:

$$\text{whenever } \frac{(f(\hat{x}^k) + h_{\hat{x}}^k) - \overline{(f+h)}_a(\hat{x}^k)}{t_k |G^k + N^k|^2} \leq -\beta \text{ set } t_k = 10t_k \text{ and loop to Step 1,} \quad (12)$$

for a parameter $\beta \in (0, 1)$ chosen at the initialization step. By making use of the above simple rule, and preventing t_k from decreasing before getting a “good candidate” x^{k+1} , the proximal bundle algorithm remains convergent. Clearly, convergence is ensured up to the oracle accuracy: the “optimal” value found by the algorithm has an error not larger than the precision $\eta_h + \eta_+$. The *partly asymptotically exact* variants let the accuracy vary with the iterates in a manner that drives $\eta_h^k + \eta_+^k$ to 0 at serious steps, hence yielding exact solutions in the limit; we refer to [44] for further information.

It is often the case that the QP subproblem solution takes a significant proportion of the total computational time. The aggregate linearization (11) plays an important role for controlling the QP size and making iterations less time consuming. Proximal bundle methods, and some level variants with restricted memory, work with models in (5) whose index set J_k can be *compressed* in Step 5 of the Algorithm Pattern 1. More precisely, for convergence purposes, it is enough to include in the new bundle set J_{k+1} the linearization of the last iterate and the aggregate linearization (11). Nevertheless, the *speed* of convergence may be impaired if the bundle is too economical; in this case, the selection mechanism may offer a better trade-off between number of iterations to reach convergence and the time spent in each individual iteration.

3.2 Level Bundle Methods

The level bundle method was proposed in [30] for convex unconstrained and constrained optimization problems, saddle-point problems, and variational inequalities. When compared to the proximal family, the level class can be considered as making *primal-dual* approximations, in the following sense. Start by rewriting the proximal point operator equivalence, multiplying the minimand therein by the positive factor t :

$$\bar{x} \text{ solves (1a)} \iff \bar{x} \text{ solves } \min_{x \in X} t(f(x) + h(x)) + \frac{1}{2}|x - \bar{x}|_M^2.$$

Then, interpreting this factor as a *multiplier* gives the relation

$$\bar{x} \text{ solves (1a)} \iff \bar{x} \text{ solves } \begin{cases} \min_{x \in X} & \frac{1}{2}|x - \bar{x}|_M^2 \\ \text{s.t.} & f(x) + h(x) \leq f(\bar{x}) + h(\bar{x}). \end{cases}$$

Once again, to make the equivalence above implementable, methods in the level variant replace \bar{x} by the current stability center, $f + h$ by some (aggregate or disaggregate) model, and the right

hand side term in the constraint by a *level* parameter. Also, the M -norm is extended to general stability functions derived from a smooth strongly convex function $\omega : \mathfrak{R}^n \rightarrow \mathfrak{R}_+$ that can yield a subproblem that is conic and no longer quadratic. Typically, the parameter set in the level variant is

$$\text{pars} = \left\{ \begin{array}{ll} \text{model} & \left\{ \begin{array}{l} \widetilde{(f+h)}_k \quad \text{if aggregate variant} \\ \check{f}_k + \check{h}_k \quad \text{if disaggregate variant} \end{array} \right. \\ \text{stability center} & \hat{x}^k \in \mathcal{X}, \\ \text{stability function} & \omega(x) - \langle \nabla \omega(\hat{x}^k), x - \hat{x}^k \rangle, \\ \text{level parameter} & f_{\text{lev}}^k \in (f_{\text{low}}^k, f_{\text{up}}^k), \text{ for} \\ \text{lower and upper bounds} & f_{\text{low}}^k \leq \nu_{\text{opt}} \text{ and } f_{\text{up}}^k \geq \nu_{\text{opt}}. \end{array} \right.$$

As mentioned in [6], the function ω can be chosen to exploit the geometry of the feasible set \mathcal{X} (that may be non-polyhedral), to get nearly dimension-independent complexity estimates of the maximum number of iterations necessary to reach an optimality gap $f_{\text{up}}^k - f_{\text{low}}^k$ inferior to a given tolerance.

Complexity estimates are obtained if the feasible set \mathcal{X} is compact, an assumption present in most level bundle methods, [30, 26, 12, 6, 28, 47]. Level bundle methods able to deal with unbounded feasible sets have an asymptotic analysis instead; we refer to [8, 43, 5, 35] and [12, 43, 35] for variants handling exact and inexact oracles, respectively.

Subproblem Definition

In this case, subproblem (6) has the form

$$x^{k+1} = \arg \min_{x \in \mathcal{X}} \omega(x) - \langle \nabla \omega(\hat{x}^k), x - \hat{x}^k \rangle \quad \text{s.t.} \quad \widetilde{(f+h)}_k(x) \leq f_{\text{lev}}^k. \tag{13}$$

For the (canonical) stability function $\omega(x) = \frac{1}{2} \langle x, x \rangle$ in particular, subproblem (13) is the quadratic program

$$x^{k+1} = \left\{ \begin{array}{ll} \arg \min & \frac{1}{2} |x - \hat{x}^k|^2 \\ \text{s.t.} & \widetilde{(f+h)}_k(x) \leq f_{\text{lev}}^k \\ & x \in \mathcal{X}, \end{array} \right.$$

corresponding to the orthogonal projection of the stability center \hat{x}^k onto the polyhedron $\mathcal{X} \cap \mathcal{X}_{\text{lev}}^k$, for

$$\mathcal{X}_{\text{lev}}^k := \{x \in \mathfrak{R}^n : \widetilde{(f+h)}_k(x) \leq f_{\text{lev}}^k\}, \tag{14}$$

a nonempty set, since the level parameter $f_{\text{lev}}^k \geq \min_{x \in \mathcal{X}} \widetilde{(f+h)}_k(x)$.

Similarly to the proximal variant, the optimality conditions for the QP subproblem above give an iterate x^{k+1} as in (8), only that now t_k is replaced by $\mu_k = \sum_{j \in \mathcal{J}_k} \alpha_j^k$ and

$$G^k = \frac{1}{\mu_k} \sum_{j \in \mathcal{J}_k} \alpha_j^k (g^j + s^j) \in \partial \widetilde{(f+h)}_k(x^{k+1})$$

for a Lagrange multiplier α^k that is no longer simplicial but conical and satisfies the following relations for all $j \in J_k$

$$\sum_{j \in J_k} \alpha_j^k \geq 1, \quad \alpha_j^k \geq 0, \quad \alpha_j^k [\overline{(f+h)}_j(x^{k+1}) - f_{lev}^k] = 0,$$

Unlike the proximal bundle variant, in the level family the dual variables of problem (13) can be unbounded.

Stability Center and Optimality Certificate

The rule to update the stability center \hat{x}^k is more flexible than in the proximal family. For exact oracles, the initial article [30] sets $\hat{x}^k = x^k$ for all k and, hence, does not have the *restricted memory* feature: with this approach compression is not possible. The proximal level variant [26] updates the center using the proximal bundle method rule (9) and, as such, works with restricted memory; similarly for [8] and [35]. Finally, [5] keeps the stability center fixed along the whole iterative process by setting $\hat{x}^k = \hat{x}^1$ for all k .

The optimality certificate checks the gap between the upper and lower bounds that also define the level parameter

$$\left. \begin{aligned} f_{low}^k &:= \min_{x \in \mathcal{X}} \overline{(f+h)}_k(x). \\ f_{up}^k &:= \min_{j \leq k} \{f(x^j) + h_x^j\} \end{aligned} \right\} \Rightarrow f_{lev}^k := \gamma f_{low}^k + (1 - \gamma) f_{up}^k,$$

for some coefficient $\gamma \in (0, 1)$. For polyhedral feasible sets \mathcal{X} , the computation of the lower bound amounts to solving an LP; for more economical alternative definitions of the lower bound, we refer to [43].

When the feasible set \mathcal{X} is compact, the stopping test in Step 2 of Algorithmic Pattern 1 checks if the optimality gap $f_{up}^k - f_{low}^k$ is below some tolerance. For the unbounded setting the stopping test of the proximal bundle method must be considered as well: stop either when the optimality gap is sufficiently small, or when both $|G^k + N^k|$ and ϕ_k are small.

Impact of Inexactness

It was shown in [41] that if the feasible set is compact, no additional procedure needs to be done to cope with inexactness from the oracle. The non-compact setting is more intricate. Basically, the level parameter remains fixed when the oracle error is deemed too large, using the test (12) to detect large “noise”. For more information on how to deal with oracle errors in level bundle methods when in (1a) the feasible set is unbounded see [35].

3.3 Doubly Stabilized Bundle Methods

It is often observed empirically that generally proximal bundle methods perform better than the level variants when $\mathcal{X} = \mathfrak{R}^n$ in problem (1a). On the other hand, the situation is reversed when

the feasible set is a compact polyhedron. The *doubly stabilized* bundle method [46] combines in a single algorithm the main features of the proximal and level methods, in an effort to exploit the best features of both variants.

A possible parameter set in this variant is

$$\text{pars} = \begin{cases} \text{model} & \begin{cases} \overline{(f+h)}_k & \text{if aggregate variant} \\ \check{f}_k + \check{h}_k & \text{if disaggregate variant} \end{cases} \\ \text{stability center} & \hat{x}^k \in \{x^k\}, \\ \text{stability function} & |\cdot|, \\ \text{prox-stepsizes} & t_k > 0, \\ \text{level parameter} & f_{\text{lev}}^k \in (f_{\text{low}}^k, f_{\text{up}}^k), \text{ for} \\ \text{lower and upper bounds} & f_{\text{low}}^k \leq \nu_{\text{opt}} \text{ and } f_{\text{up}}^k \geq \nu_{\text{opt}}. \end{cases}$$

Subproblem Definition

Letting $i_{\mathcal{X}_{\text{lev}}^k}(\cdot)$ denote the indicator function of the level set (14), the objective function of subproblem (6) is given by

$$O_k(x; \text{pars}) = \overline{(f+h)}_k(x) + \frac{1}{2t_k} |x - \hat{x}^k|^2 + i_{\mathcal{X}_{\text{lev}}^k}(x).$$

As long as the level set $\mathcal{X}_{\text{lev}}^k$ is nonempty, the new iterate is given by the following particularization of subproblem (6):

$$x^{k+1} = \arg \min_{x \in \mathcal{X}} \overline{(f+h)}_k(x) + \frac{1}{2t_k} |x - \hat{x}^k|^2 + i_{\mathcal{X}_{\text{lev}}^k}(x), \tag{15}$$

which can be rewritten as

$$x^{k+1} = \begin{cases} \arg \min & \overline{(f+h)}_k(x) + \frac{1}{2t_k} |x - \hat{x}^k|^2 \\ \text{s.t.} & \overline{(f+h)}_k(x) \leq f_{\text{lev}}^k \\ & x \in \mathcal{X}. \end{cases}$$

Lemma 2 in [46] shows that the solution x^{k+1} of the QP (15) solves either (7) or (13) (when $\omega(x) = \frac{1}{2}\langle x, x \rangle$ in the latter problem), so the doubly stabilized variant indeed combines the proximal and level methods. The numerical experiments reported in [46] show a gain in performance, likely due to the increased versatility of this variant.

The unique QP solution is characterized as follows; see [46, Prop. 2.1]:

$$x^{k+1} = \hat{x}^k - t_k \mu_k (G^k + N^k) \text{ with } \begin{cases} G^k & := \frac{1}{\mu_k} \sum_{j \in J_k} \alpha_j^k (g^j + s^j) \in \partial \overline{(f+h)}_k(x^{k+1}) \\ N^k & := -\frac{x^{k+1} - \hat{x}^k}{t_k \mu_k} - G^k \in \partial i_{\mathcal{X}}(x^{k+1}). \end{cases}$$

The multipliers $\alpha^k \geq 0$ and $\mu_k \geq 1$ satisfy the following relations for all $j \in \mathcal{J}_k$

$$\begin{aligned} \mu_k &= \sum_{j \in \mathcal{J}_k} \alpha_j^k, & \lambda_k &= \mu_k - 1 \\ \alpha_j^k [\widetilde{(f+h)}_k(x^{k+1}) - \overline{(f+h)}_j(x^{k+1})] &= 0, & \lambda_k [\widetilde{(f+h)}_k(x^{k+1}) - f_{lev}^k] &= 0. \end{aligned}$$

As in the previous methods, the α_j^k -multipliers can be employed to select active linearizations. With this variant, the compression mechanism is also applicable.

Stability Center and Optimality Certificate

The rule to update the stability center \hat{x}^k is (9), as in the proximal bundle method. The stopping test combines the proximal and level criteria: the algorithm stops when either $|G^k + N^k|$ and ϕ_k are small, or when the optimality gap is close to zero.

An additional bonus is that the μ_k -multipliers can be used to update the prox-stepsize. Specifically, in view of the optimality conditions above, given some $\gamma \in (0, 1)$, the rule

$$\text{sets } t_{k+1} = \gamma t_k \text{ in case of null step and } t_{k+1} = \mu_k t_k \text{ if } x^{k+1} \text{ is a serious iterate.}$$

Impact of Inexactness

As shown in [46], a distinctive feature of this variant is that it can be employed with both exact or inexact oracles without any changes, as long as the last “proximal step” linearization is kept in the information bundle.

Having laid down the background for problems with simple constraints, we now turn our attention to the constrained problem (1b).

4 CONSTRAINED SETTING

In spite of having only one scalar constraint, formulation (1b) is general and covers problems with m scalar convex constraints $h_j(x) \leq 0, j = 1, \dots, m$, just by taking

$$h(x) := \max_{j=1, \dots, m} h_j(x).$$

A key definition in NCO is the so-called *improvement function*, a merit function which transforms (1b) into an unconstrained problem. The potential interest of such reformulation is to make it possible for algorithms to work without forcing feasibility. Indeed, constrained bundle methods such as those in [36] and [21, Ch. 5] can only work with feasible points. Although such a setting can be useful in some cases, there are also some applications for which the problem of computing one feasible point is as difficult as to solve the optimization problem itself (an example is the hydroreservoir management in [3, 2] mentioned in Section 5 below). For such applications, infeasible methods are of foremost importance. Using exact penalty objective functions as in [20, 22] is one possibility, keeping in mind that estimating a suitable value of the

penalty parameter is sometimes a delicate task. In what follows we focus on methods developed under a different paradigm, making use of improvement functions. For another alternative, we refer to the filter method in [18].

4.1 Improvement Function

When the optimal value v_{opt} of problem (1b) is known, a direct reformulation of (1b) consists in minimizing the function $\max\{f(x) - v_{\text{opt}}, h(x)\}$ over the set \mathcal{X} . Since often the optimal value is not available, instead one considers approximations of the form

$$F(x; \tau_1^k, \tau_2^k) := \max\{f(x) - \tau_1^k, h(x) - \tau_2^k\} \quad \text{for targets } \begin{cases} \tau_1^k \approx v_{\text{opt}} \\ \tau_2^k \approx 0, \end{cases} \quad (16)$$

which will be in turn modeled by cutting-plane functions.

The *improvement* function (16) finds a compromise between optimality and feasibility, represented by the first and second terms in the max-operation, respectively. The target choice depends on the method; [30] takes $\tau_1^k := f_{\text{low}}^k$, a lower bound for v_{opt} , and sets $\tau_2^k := 0$; in [23], the first target is the sum of the objective function value at the current stability center and a weighted measure of its feasibility; and other rules are considered in [23, 51, 3].

As for the unconstrained case, we revise different bundle variants suited to (1b), starting with the proximal family.

4.2 Constrained Proximal Bundle Methods

The method introduced in [51] takes the current serious step function value for the first target and $\tau_2^k = 0$ and applies an unconstrained proximal bundle method to the function

$$F_k(x) := \max \{ f(x) - f(\hat{x}^k), h(x) \}.$$

When the next serious iterate is generated, the first target is replaced accordingly, making the necessary changes to the bundle. These changes are not straightforward because, as explained in [51], ensuring descent for the improvement function above does not necessary entail descent for the objective in (1b).

The parameters characterizing this family of methods is

$$\text{pars} = \begin{cases} \text{model} & \Phi_k(x) := \max\{\check{f}(x) - f(\hat{x}^k), \check{h}(x)\}, \\ \text{stability center} & \hat{x}^k \in \{x^k\}, \\ \text{stability function} & |\cdot| \text{ the Euclidean norm,} \\ \text{prox-stepsize} & t_k > 0. \end{cases}$$

Subproblem Definition

In this variant, the objective function in (6) is

$$O_k(x; \text{pars}) = \Phi_k(x) + \frac{1}{2t_k} |x - \hat{x}^k|^2,$$

yielding a QP problem, with unique solution as in (8) written with $\partial(\widetilde{f+h})_k(x^{k+1})$ therein replaced by

$$\partial\Phi_k(x^{k+1}) = \begin{cases} \partial\check{f}(x^{k+1}) & \text{if } \check{f}(x^{k+1}) - f(\hat{x}^k) > \check{h}(x^{k+1}), \\ \text{conv}\{\partial\check{f}(x^{k+1}) \cup \partial\check{h}(x^{k+1})\} & \text{if } \check{f}(x^{k+1}) - f(\hat{x}^k) = \check{h}(x^{k+1}), \\ \partial\check{h}(x^{k+1}) & \text{if } \check{f}(x^{k+1}) - f(\hat{x}^k) < \check{h}(x^{k+1}). \end{cases}$$

Stability Center and Optimality Certificate

The stability center, which also defines the first target, is changed when the current improvement function is sufficiently reduced using a test akin to (9), i.e.,

$$F_k(x^{k+1}) \leq F_k(\hat{x}^k) - m(F_k(\hat{x}^k) - \Phi_k(x^{k+1})).$$

Because the improvement function is a max-function, the test requires progress on *both* optimality and feasibility criteria. Note also that, as $F_k(\hat{x}^k) = \max\{0, h(\hat{x}^k)\}$, when the center is infeasible it is possible that $f(\hat{x}^{k+1}) > f(\hat{x}^k)$. So, outside the feasible region, the method is not monotone with respect to f but it is monotone with respect to h , thus decreasing infeasibility. The situation is reversed once a feasible stability center is found.

The non-straightforward changes in the bundle alluded above refer to the fact that when $\hat{x}^{k+1} = x^{k+1}$, the bundle that was used to model the function F_k must now be adapted to model the new improvement function $F_{k+1}(x) = \max\{f(x) - f(x^{k+1}), h(x)\}$. We refer to Section 3 in [51] for details.

Approximate optimality is declared when the center satisfies the inclusion $G^k + N^k \in \partial_{\epsilon_k} F_k(\hat{x}^k)$ with both $|G^k + N^k|$ and ϵ_k sufficiently small. When calculations are exact or for inexact oracles with bounded \mathcal{X} , this test boils to the one using ϕ_k from (10).

Impact of Inexactness

Suppose now the constraint h in (1b) is hard to compute. The ideas above were generalized to handle inexact oracles in [3], using a test similar to (12) *mutatis mutandis*. The extension addresses two important issues, described below.

- The targets in the improvement function are more general:

$$\tau_1^k = f(\hat{x}^k) + \rho_k \max\{h_{\hat{x}}^k, 0\} \quad \text{and} \quad \tau_2^k = \sigma_k \max\{h_{\hat{x}}^k, 0\}$$

for parameters $\rho_k \in [0, \infty]$ and $\sigma_k \in [0, 1]$ such that $1 - \sigma_k + \rho_k$ is bounded away from zero. The function F_k from (16) is recovered by taking $\sigma_k = \rho_k = 0$ at all iterations, a setting that may slow down the convergence when getting close to the boundary of the feasible set, due to zigzagging. Taking $\rho_k > 0$ prevents oscillating when close to a solution and may be beneficial for the speed of convergence. Note also that the target choice also allows for driving ρ_k to infinity, as in an exact penalty approach.

- Rather than requiring descent for the improvement function, the stability center is updated when *either* the center is feasible and the new point is feasible and gives descent for the objective function, *or* the center is infeasible and the new point reduces infeasibility. This weaker condition is likely to update serious steps more often.

4.3 Constrained Bundle Methods in the Level Family

The first constrained level method able to deal with inexactness was proposed by [12], assuming the oracle is of the lower type with vanishing errors, that is $\eta_h^k \rightarrow 0$. Further improvements, incorporating the on-demand accuracy setting from [43], were considered in [13]. The basic idea in these articles is to apply some unconstrained level bundle method to solve the linearly constrained problem $\min_{x \in \mathcal{X}} F_k(x)$ where

$$F_k(x) := F(x; f_{\text{low}}^k, 0) \text{ from (16) and with } f_{\text{low}}^k \uparrow \nu_{\text{opt}}.$$

Since the algorithms in [30, 26, 12, 13] make use of certain dual variables to update the level parameter for the cutting-plane model of F_k , below we grouped them in the family of *primal-dual* level bundle methods. In contrast, the methods proposed in [2] do not set the objective function in the subproblems to be F_k but rather use the improvement function as a measure to certify optimality. The authors justify this choice as being more natural when dealing with inexactness from the oracle. The method was extended in [42] to solve convex mixed-integer non-linear programming problems.

For both types of methods, the parameter set is

$$\text{pars} = \begin{cases} \text{models} & \begin{cases} \check{f}_k & \text{for } f \\ \check{h}_k & \text{for } h \end{cases} \\ \text{stability center} & \hat{x}^k \in \mathcal{X}, \\ \text{stability function} & |\cdot|, \\ \text{lower bound} & f_{\text{low}}^k \leq \nu_{\text{opt}}, \\ \text{level-parameter} & f_{\text{lev}}^k \geq f_{\text{low}}^k, \\ \text{improvement function} & F_k. \end{cases}$$

4.3.1 Primal-Dual Constrained Level Bundle Methods

The algorithms in [30, 12] define the objective function in (6) as

$$O_k(x; \text{pars}) = \frac{1}{2} |x - \hat{x}^k|^2 + i_{\mathcal{X}_{\text{lev}}^k}(x).$$

In these primal-dual variants, the level set $\mathcal{X}_{\text{lev}}^k$ depends on a dual variable $\alpha_k \in (0, 1)$, computed by solving a one-dimensional optimization problem related to the max-operation defining F_k ,

so that $\alpha_k \check{f}_k(x) + (1 - \alpha_k) \check{h}_k(x)$ is a cutting-plane approximation of the improvement function. Accordingly, the iterate x^{k+1} solves the QP subproblem:

$$x^{k+1} = \begin{cases} \arg \min & \frac{1}{2} |x - \hat{x}^k|^2 \\ \text{s.t.} & \alpha_k \check{f}_k(x) + (1 - \alpha_k) \check{h}_k(x) \leq f_{\text{lev}}^k \\ & x \in \mathcal{X}. \end{cases}$$

The level parameter f_{lev}^k is updated by solving an LP, as in the unconstrained variant.

Stability Center and Optimality Certificate

The rule proposed in [30] to update the stability center is simple: just take \hat{x}^k as the current iterate x^k . Differently, [26] updates \hat{x}^k only after observing enough progress in the improvement function F_k (as for the proximal variants). The new stability center can be any point in the information bundle (for example, a feasible iterate with lowest objective function value). The choice $\hat{x}^k = x^1$ for all k is also possible for the method proposed in [26].

The optimality certificate is given by the improvement function: the algorithm stops with a δ_{Tol} -solution when $\min_{j \leq k} F_k(x^j) \leq \delta_{\text{Tol}}$.

Impact of Inexactness

Publication [12] works with lower oracles, so $\eta_+ \equiv 0$, and the algorithm assumes the oracle error η_h^k is driven to zero along the iterative process: eventually the method coincides with the variant for exact oracles.

4.3.2 Constrained Level Bundle Methods

The method proposed in [2] defines the objective function in (6) as

$$O_k(x; \text{pars}) = \frac{1}{2} |x - \hat{x}^k|^2 + i_{\mathcal{X}_{\text{lev}}^k}(x),$$

where the level set $\mathcal{X}_{\text{lev}}^k$ is the intersection of the level set for the objective model \check{f}_k with an outer approximation of the constraint set. Accordingly, the next iterate solves the QP subproblem:

$$x^{k+1} = \begin{cases} \arg \min & \frac{1}{2} |x - \hat{x}^k|^2 \\ \text{s.t.} & \check{f}_k(x) \leq f_{\text{lev}}^k \\ & \check{h}_k(x) \leq 0 \\ & x \in \mathcal{X}. \end{cases} \tag{17}$$

Lemma 2 in [2] assures that f_{lev}^k is a lower bound for the optimal value v_{opt} of (1b) whenever the feasible set of QP (17) is empty.

The rule given in [2] to update the lower bound and the level parameter does not rely on dual variables, and is much simpler:

$$f_{\text{lev}}^k = f_{\text{low}}^k + \gamma \min_{j \leq k} F_k(x^j)$$

$$f_{\text{low}}^k = \begin{cases} f_{\text{lev}}^k & \text{if } \mathcal{X}_{\text{lev}}^k \text{ is empty} \\ f_{\text{low}}^{k-1} & \text{otherwise.} \end{cases}$$

Different from the primal-dual family, no additional optimization subproblem needs to be solved at each iteration (neither to define the dual variable α^k nor to obtain f_{low}^k). In this method, both selection and compression of the bundle is possible.

Stability Center and Optimality Certificate

Similar to the unconstrained level bundle method given in [43], a new stability center \hat{x}^k is chosen in the information bundle whenever $\mathcal{X}_{\text{lev}}^k$ is empty. The stopping test is identical to the one in [30, 26, 12], checking that $\min_{j \leq k} F_k(x^j)$ is sufficiently small.

Impact of Inexactness

The algorithm is robust to noise, in the sense that no additional test is required to cope with inexactness. With an exact oracle, the improvement function is always nonnegative. Negative values for $F_k(\cdot)$ may arise when the oracle is inexact, in which case the method terminates with an $(\eta_h + \eta_+ + \delta_{\text{Tol}})$ -solution to problem (1b). For lower oracles, driving to zero the error η_h^k for certain iterates named *substantial* ensures that asymptotically an exact solution is found; [2].

5 ONE PROBLEM, TWO FORMULATIONS

When dealing with real-life problems, *modeling* is an issue as important as *solving* the resulting optimization problem. As an illustration, we now consider an example from hydroreservoir management and represent the same physical problem in two different manners, yielding either a problem of the form (1a) or of the form (1b).

Figure 1 represents a set of power plants along a hydrological basin in which the reservoirs uphill have a volume subject to random streamflows due to melted snow. In the figure, there are N reservoirs, each one with volume $V^i(t)$ for $i = 1, \dots, N$ and a certain time step t , out of which the first r are subject to uncertain inflows. For the full model details, we refer to [1].

Since power plants are cascaded along the same basin, water released uphill fills the reservoirs downhill. The increased volume allows the plants downhill either to produce hydropower by directing water through turbines or to release water that is pumped to refill uphill reservoirs. These operations make the plants interdependent and, as such, the best practice is to manage them together rather than individually.

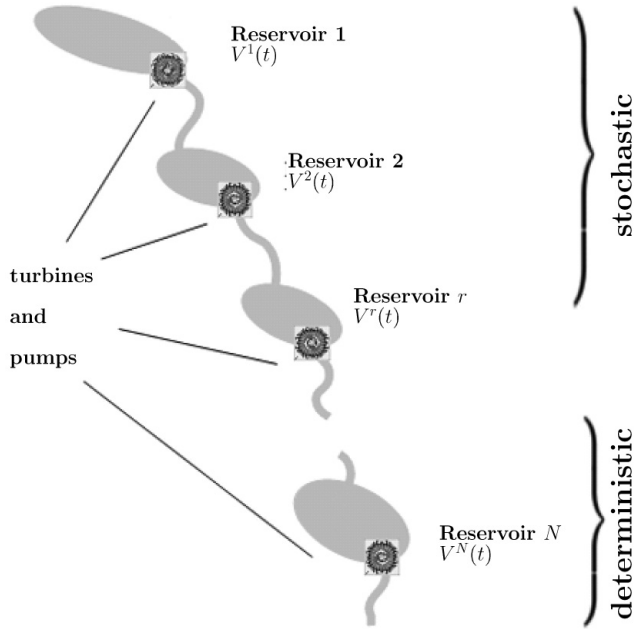


Figure 1 – Schematic Representation of a Hydrovalley.

In particular, an important concern is to keep the reservoirs above some *min-zone* constraints, i.e., lower bounds set by the system operator for navigation or irrigation purposes. Since the reservoir volumes are uncertain, so are the min-zones values, and the optimization problem that defines the remuneration for each power plant has a stochastic nature. A possibility to define a turbine throughput and pumping strategy that is suitable for the whole hydrovalley is to endow the optimization problem with *probabilistic constraints*.

Suppose a (random) unit price π , a (random) lower bound ℓ , and a probability level $p \in (0, 1)$ are given. Letting $x \in \mathbb{R}^n$ denote the abstract vector gathering the decision variables (release water through the turbines, pumping, etc), the hydroreservoir management problem has the form

$$\begin{cases} \max & \langle \pi, x \rangle \\ \text{s.t.} & x \in \mathcal{X} \text{ a bounded convex polyhedron,} \\ & \mathbb{P}(\ell \leq x) \geq p. \end{cases}$$

When streamflows are represented by some autoregressive model with Gaussian innovations, the joint chance constraints above define a convex feasible set and

$$h(x) := \log(p) - \log(\mathbb{P}(\ell \leq x)) \tag{18}$$

is a convex function, whose differentiability depends on the covariance matrix of the Gaussian noises. The oracle to compute values for h and a (sub)gradient needs to perform numerical integration in high dimensions. More precisely, since, typically, the horizon has $T = 48$ or

96 time steps, the vector x , formed by subvectors for each plant, each one of length T , has a dimension of several hundreds.

We are therefore in our initial setting: the hydroreservoir management problem fits structure (1b) with a linear $f(\cdot) := \langle \pi, \cdot \rangle$, easy to evaluate, and a difficult constraint of the form (18).

To see how to cast the hydroreservoir problem in the format (1a), we follow [52, Chapter 4.3.2] and consider the set of p -efficient points, a generalization of the p -quantile:

$$\mathcal{Z}_p := \{z \in \mathbb{R}^n : F_\ell(z) \geq p\}$$

where $F_\ell(z) = \mathbb{P}(x \geq z)$ is the distribution function of the min-zone bound ℓ .

As explained in [52, Chapter 4.3.3], the equivalent reformulation of the hydroreservoir problem

$$\begin{cases} \max & \langle \pi, x \rangle \\ \text{s.t.} & x \in \mathcal{X} \\ & x \geq z \\ & z \in \mathcal{Z}_p \end{cases}$$

can be dualized by introducing Lagrange multipliers $0 \leq u \in \mathbb{R}^n$ for the constraint $x \geq z$. The resulting dual problem

$$\begin{cases} \min & f(u) + h(u) \\ \text{s.t.} & u \geq 0 \end{cases} \quad \text{with} \quad \begin{cases} f(u) & := \max_{x \in \mathcal{X}} \langle \pi - u, x \rangle \\ h(u) & := \max_{z \in \mathcal{Z}_p} \langle u, z \rangle \end{cases}$$

fits setting (1a). Oracles for estimating the value and gradient of the hard function h , involving integer programming, are considered in [10].

The loop is looped: starting with a problem as in (1b) we found an equivalent problem in the form (1a). The decision of which formulation to choose to eventually solve the hydroreservoir management problem will depend on the availability of the oracle and NCO solvers. For instance, designing an oracle for (18) requires having a good code for numerical integration while the hard function depending on p -efficient points calls for combinatorial optimization techniques. A final, not less important, consideration is to analyze what is the desired *output* of the problem (only the optimal value, or a solution, or a feasible point, etc).

6 FINAL COMMENTS

We gave a panorama of state-of-the-art bundle algorithms that were developed in the last 10-15 years to deal with inaccurate oracle information. Applications in energy, finance, transportation or planning often require solving with high precision and in a reliable manner large-scale problems for which the oracle information is hard, if not impossible, to obtain. The new generation of bundle methods reviewed in this work constitutes an important set of optimization tools for such real-life problems.

Our presentation covers methods for unconstrained, linearly constrained and nonlinearly constrained convex nonsmooth optimization problems. As made clear by the hydroreservoir example in Section 5, the choice of which specific formulation and bundle variant to employ depends

largely on the *structure* of the problem to be solved. For instance, for large-scale unconstrained optimization problems the proximal bundle method seems to be more suitable than the level one, since each iterate can be determined by maximizing a quadratic function over a simplex (this is a QP subproblem, dual to (7)). On the other hand, if the feasible set is bounded, the level bundle variant might be preferable because the fact of having a lower bound allows the method to build an optimality gap, a stopping criterion that is easier to satisfy in general. Similarly if the optimal value of the problem is known in advance. The doubly stabilized variant can be a good solver for repeated solution of different types of problems, since it gives an automatic mechanism to combine the advantages of the proximal and level methods for the unconstrained or linearly constrained setting.

Notwithstanding, each bundle variant has parameters to be dealt with, such as the prox-stepsize or the level parameter. The convergence theory for each method relies on the proper updating of such parameters. For the proximal family, for instance, the stepsize cannot increase at null steps and (as for subgradient methods) must form a divergent series. For the level family, f_{lev}^k is often a convex combination of the current upper and lower bounds. Depending on the updating rule for the lower bound, the level parameter can make the subproblem infeasible. As long as the subproblem is a QP, this is not a major concern: most of modern QP solvers return with an exit flag informing infeasibility. In this case, the lower bound and level parameter are easily updated, see Subsection 4.3.2. Furthermore, to prevent the level set from being empty, the lower bound can be computed by solving an LP at each iteration, as discussed in Subsection 3.2.

An important point to note is that for lower and on-demand accuracy oracles (not for upper oracles), inexact bundle methods can eventually find an exact solution even if the oracle information is inexact most of time: the solution quality depends only on how accurate the oracle is at serious steps.

Regarding computational packages, several of the variants described above are freely available for academic use upon request to the authors, including [15] and [44].

We finish by commenting on several future directions of research.

For the proximal family, [4] makes a first incursion into new criteria to define stability centers. Differently from (9), the serious sequence may have non-monotone function values. The idea is to have more serious steps and to avoid the algorithm stalling at null steps. It would be interesting to see if it is possible to relate these new criteria to crucial objects in the level family, such as the optimality gap.

Concerning research impacting applications, we can mention four topics. First, asynchronous parallelization as in [14] deserves further investigation. Second, the issue of primal recovery when solving a problem via Lagrangian relaxation: it remains to understand how the primal solutions are polluted by “noise” when the Lagrangian subproblems are solved inexactly and some inexact bundle algorithm is used. Third, regarding mixed-integer problems, the extension of bundle methods [42] left open some issues on complexity and restricted memory. Fourth, when in finance or economics there are multiple solutions, some special one can be singled out

by, for instance, looking for the solution that is closest to some reference point. Exploiting the structure in such nonsmooth minimal norm solution problems could lead to the design of new specialized solution algorithms, based on the bundle methodology.

Finally, along the lines of the work [38] for proximal bundle methods, it would be interesting to incorporate $\mathcal{V}\mathcal{U}$ -decomposition into the level family to devise superlinearly convergent level bundle methods.

ACKNOWLEDGMENTS

We are grateful to Robert Mifflin for his careful reading and useful remarks. The second author is partially supported by Grants CNPq 303840/2011-0, AFOSR FA9950-11-1-0139, as well as by PRONEX-Optimization and FAPERJ.

REFERENCES

- [1] VAN ACKOOIJ W, HENRION R, MÖLLER A & ZORGATI R. 2013. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, accepted for publication.
- [2] VAN ACKOOIJ W & DE OLIVEIRA W. 2014. Level bundle methods for constrained convex optimization with various oracles. *Computational Optimization and Applications*, **57**: 555–597.
- [3] VAN ACKOOIJ W & SAGASTIZÁBAL C. 2014. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM Journal on Optimization*, **24**(2): 733–765. Doi=10.1137/120903099.
- [4] ASTORINO A, FRANGIONI A, FUDULI A & GORGONE E. 2013. A nonmonotone proximal bundle method with (potentially) continuous step decisions. *SIAM Journal on Optimization*, **23**: 1784–1809.
- [5] BELLO CRUZ JY & DE OLIVEIRA W. 2014. Level bundle-like algorithms for convex optimization. *Journal of Global Optimization*, **59**(4): 787–809. Doi=10.1007/s10898-013-0096-4.
- [6] BEN-TAL A & NEMIROVSKI A. 2005. Non-euclidean restricted memory level method for large-scale convex optimization. *Math. Program.*, **102**: 407–456.
- [7] BORGHETTI A, FRANGIONI A, LACALANDRA F & NUCCI C. 2003. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, **18**: 313–323.
- [8] BRANNLUND U, KIWIEL KC & LINDBERG PO. 1995. A descent proximal level bundle method for convex nondifferentiable optimization. *Operations Research Letters*, **17**: 121–126.
- [9] CHENEY E & GOLDSTEIN A. 1959. *Newton's method for convex programming and Tchebycheff approximations*, **1**: 253–268.
- [10] DENTCHEVA D & MARTINEZ G. 2003. Regularization methods for optimization problems with probabilistic constraints. *Math. Program.*, **138**: 223–251.
- [11] EMIEL G & SAGASTIZÁBAL C. 2010. Incremental-like bundle methods with application to energy planning. *Computational Optimization and Applications*, **46**: 305–332.
- [12] FÁBIÁN C. 2000. Bundle-type methods for inexact data. *Central European Journal of Operations Research*, **8** (special issue, T. Csendes and T. Rapcsk, eds.), pp. 35–55.

- [13] FÁBIÁN C. 2013. *Computational aspects of risk-averse optimisation in two-stage stochastic models*, tech. rep., Institute of Informatics, Kecskemét College, Hungary. Available at: http://www.optimization-online.org/DB_HTML/2012/08/3574.html.
- [14] FISCHER F & HELMBERG C. 2014. A parallel bundle framework for asynchronous subspace optimization of nonsmooth convex functions. *SIAM Journal on Optimization*, **24**(2): 795–822. Doi=10.1137/120865987.
- [15] FRANGIONI A. 2002. Generalized bundle methods. *SIAM Journal on Optimization*, **13**: 117–156.
- [16] FRANGIONI A & GENDRON B. 2013. A stabilized structured Dantzig Wolfe decomposition method. *Math. Program.*, **140**: 45–76.
- [17] HINTERMÜLLER M. 2001. A proximal bundle method based on approximate subgradients. *Computational Optimization and Applications*, **20**: 245–266.
- [18] KARAS E, RIBEIRO A, SAGASTIZÁBAL C & SOLODOV M. 2009. A bundle-filter method for nonsmooth convex constrained optimization. *Math. Program.*, **116**: 297–320.
- [19] KELLEY J. 1960. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, **8**: 703–712.
- [20] KIWIŁ K. 1985. An exact penalty function algorithm for nonsmooth convex constrained minimization problems. *IMA J. Numer. Anal.*, **5**: 111–119.
- [21] KIWIŁ K. 1985. *Methods of descent for nondifferentiable optimization*. Springer-Verlag, Berlin.
- [22] KIWIŁ K. 1991. Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. *Math. Programming*, **52**: 285–302.
- [23] KIWIŁ K. 2008. A method of centers with approximate subgradient linearizations for nonsmooth convex optimization. *SIAM Journal on Optimization*, **18**: 1467–1489.
- [24] KIWIŁ K & LEMARÉCHAL C. 2009. An inexact bundle variant suited to column generation. *Math. Program.*, **118**: 177–206.
- [25] KIWIŁ KC. 1995. Approximations in proximal bundle methods and decomposition of convex programs. *Journal of Optimization Theory and Applications*, **84**: 529–548. 10.1007/BF02191984.
- [26] KIWIŁ KC. 1995. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Math. Program.*, **69**: 89–109.
- [27] KIWIŁ KC. 2006. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, **16**: 1007–1023.
- [28] LAN G. 2013. Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization. *Math. Program.*, pp. 1–45.
- [29] LEMARÉCHAL C. 1975. An extension of Davidon methods to nondifferentiable problems. *Math. Programming Stud.*, **3**: 95–109.
- [30] LEMARÉCHAL C, NEMIROVSKII A & NESTEROV Y. 1995. New variants of bundle methods. *Math. Program.*, **69**: 111–147.
- [31] LEMARÉCHAL C, OUOROU A & PETROU G. 2011. A bundle-type algorithm for routing in telecommunication data networks. *Comput. Optim. Appl.*, **44**: 385–409.

- [32] LEMARÉCHAL C AND SAGASTIZÁBAL C. 1994. An approach to variable metric bundle methods. *Lecture Notes in Control and Information Science*, **197**: 144–162.
- [33] LEMARÉCHAL C & SAGASTIZÁBAL C. 1997. Variable metric bundle methods: From conceptual to implementable forms. *Math. Program.*, **76**: 393–410. 10.1007/BF02614390.
- [34] LIN H. 2012. An inexact spectral bundle method for convex quadratic semidefinite programming. *Computational Optimization and Applications*, **53**: 45–89.
- [35] MALICK J, DE OLIVEIRA W & ZAOURAR S. 2013. *Nonsmooth optimization using uncontrolled inexact information*, tech. rep. Available at: http://www.optimization-online.org/DB_HTML/2013/05/3892.html.
- [36] MIFFLIN R. 1977. *An algorithm for constrained optimization with semismooth functions*, **2**: 191–207.
- [37] MIFFLIN R & SAGASTIZÁBAL C. 2000. On $\mathcal{V}U$ -theory for functions with primal-dual gradient structure. *Siam Journal on Optimization*, **11**: 547–571.
- [38] MIFFLIN R & SAGASTIZÁBAL C. 2005. A $\mathcal{V}U$ -algorithm for convex minimization. *Math. Program.*, **104**: 583–608.
- [39] MIFFLIN R & SAGASTIZÁBAL C. 2012. Optimization Stories, vol. Extra Volume ISMP 2012, ed. by M. Grötschel, DOCUMENTA MATHEMATICA, ch. A Science Fiction Story in Nonsmooth Optimization Originating at IIASA, p. 460.
- [40] MOREAU J. 1965. Proximité et dualité dans un espace Hilbertien. *Bull. Soc. Math. France*, **93**: 273–299.
- [41] DE OLIVEIRA W. 2011. Inexact bundle methods for stochastic optimization. PhD thesis, Federal University of Rio de Janeiro, COPPE/UFRJ, January 2011. (In Portuguese) Available at: http://www.cos.ufrj.br/index.php?option=com_publicacao&task=visualizar&cid%5B0%5D=2192.
- [42] DE OLIVEIRA W. 2014. *Regularized nonsmooth methods for solving convex MINLP problems*, tech. rep. Available at: www.oliveira.mat.br.
- [43] DE OLIVEIRA W & SAGASTIZÁBAL C. 2014. Level bundle methods for oracles with on-demand accuracy. *Optimization Methods and Software*, **29**(6): 1180–1209. Doi=10.1080/10556788.2013.871282.
- [44] DE OLIVEIRA W, SAGASTIZÁBAL C & LEMARÉCHAL C. 2014. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Mathematical Programming*, pp. 1–37. Springer Berlin Heidelberg. Doi=10.1007/s10107-014-0809-6.
- [45] DE OLIVEIRA W, SAGASTIZÁBAL C & SCHEIMBERG S. 2011. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, **21**: 517–544.
- [46] DE OLIVEIRA W & SOLODOV M. 2013. *A doubly stabilized bundle method for nonsmooth convex optimization*, tech. rep. Available at: http://www.optimization-online.org/DB_HTML/2013/04/3828.html.
- [47] RICHTÁRIK P. 2012. Approximate level method for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, **152**: 334–350.
- [48] SAGASTIZÁBAL C. 2011. Nonsmooth optimization: Thinking outside of the black box. *SIAG/OPT Views-and-News*, **22**: 2–10.

- [49] SAGASTIZÁBAL C. 2012. Divide to conquer: decomposition methods for energy optimization. *Math. Program.*, **134**: 187–222.
- [50] SAGASTIZÁBAL C. 2013. Composite proximal bundle method. *Mathematical Programming*, **140**: 189–233.
- [51] SAGASTIZÁBAL C & SOLODOV M. 2005. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM Journal on Optimization*, **16**: 146–169.
- [52] SHAPIRO A, DENTCHEVA D & RUSZCZYŃSKI A. 2009. Lectures on Stochastic Programming: Modeling and Theory. MPS-SIAM Series on Optimization, SIAM – Society for Industrial and Applied Mathematics and Math. Program. Society, Philadelphia.
- [53] SOLODOV MV. 2003. On approximations with finite precision in bundle methods for nonsmooth optimization. *Journal of Optimization Theory and Applications*, **119**: 151–165.
- [54] WOLFE P. 1975. A method of conjugate subgradients for minimizing nondifferentiable functions. *Math. Programming Stud.*, **3**: 145–173.