ORIGINAL PAPER

Solving generation expansion planning problems with environmental constraints by a bundle method

Claudia Sagastizábal · Mikhail Solodov

Abstract   We discuss the energy generation expansion planning with environmental constraints, formulated as a nonsmooth convex constrained optimization problem. To solve such problems, methods suitable for constrained nonsmooth optimization need to be employed. We describe a recently developed approach, which applies the usual unconstrained bundle techniques to a dynamically changing "improvement function". Numerical results for the generation expansion planning are reported.

1 Introduction, motivation

One of the important applications of nonsmooth optimization (NSO) methods arises from Lagrangian relaxation or Benders' decomposition (Bonnans et al. 2006). In general, to solve large-scale complex problems with separable structure, decomposition techniques replace the original problem by a sequence of reduced-dimensional

C. Sagastizábal (✉)
CEPEL, Eletrobrás Group, Electric Energy Research Center, Rio de Janeiro, Brazil
e-mail: sagastiz@impa.br

M. Solodov
IMPA, Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110,
Jardim Botânico, Rio de Janeiro, RJ, 22460-320, Brazil
e-mail: solodov@impa.br

(or easier) local problems, linked by a master program. Nonsmoothness comes into play in the master program, which often has box or polyhedral constraints. However, for some energy optimization problems, the corresponding master program is a NSO problem with general (nonlinear) convex constraints. This is the case for the particular application considered in this work, that of generation expansion planning (GEP). This type of problems, modeled as two-stage stochastic programs with recourse, determine an optimal investment plan for the construction of new power plants, while ensuring economic and reliable supply to the future electricity demand (see Bienstock and Shapiro 1988; Campodónico et al. 2003; Marcato and Sagastizábal 2007). Generally, discounted expected costs of investment and operation are minimized subject to constraints depending on uncertain data, such as future growth of electricity demand, environmental restrictions, and primary fuel costs. As explained in Sect. 4 below, if water pollution constraints are taken into account, a Benders'-like decomposition yields a master problem which is a constrained NSO problem.

Mathematically, this means solving the problem

$$\min f(x) \quad \text{s.t.} \quad c(x) \leq 0, \tag{1}$$

where $f, c : \mathbb{R}^n \to \mathbb{R}$ are convex functions, nondifferentiable at some points. We note that there is no loss of generality in formulating (1) with only one constraint: if necessary, c can be defined as the pointwise maximum of finitely many convex functions, thus covering the case of multiple inequality constraints. In the setting of nondifferentiable c, such a transformation does not interfere with any essential structural properties of the problem.

Nonsmooth optimization (NSO) problems are often relatively difficult to solve, even when they are unconstrained. Among algorithms for NSO, we mention the subgradient, cutting-plane, analytic center cutting-plane, and bundle methods. In turn, bundle methods are subdivided into different variants, such as the trust-region (Schramm and Zowe 1992), level (Lemaréchal et al. 1995; Fábián 2000), and proximal (Kiwiel 1991). In this work, we use the constrained form of proximal bundle methods, specifically the one developed in Sagastizábal and Solodov (2005).

For unconstrained problems, iterates of a proximal bundle algorithm are generated by solving a quadratic programming problem (QP). Each QP is defined by means of a cutting-plane model of the objective function, stabilized by a quadratic term centered at the best point obtained so far (which is referred to as the last descent or "serious" iterate). An important feature of proximal bundle methods is that the size of each QP can be controlled via the so-called aggregation techniques, see for instance Bonnans et al. (2006, Ch. 9). This feature can be crucial for efficient implementations.

Constrained nonsmooth problems are more complex, and only a few practical methods can be found in the literature. Convex problems with "easy" constraints (such as bound or linear constraints) can be solved either by inserting the constraints directly into each QP, or by projecting iterates onto the feasible set. For problems with nonlinear constraints, one possibility is to solve an equivalent unconstrained problem with an exact penalty objective function, e.g., Kiwiel (1991). This approach, however, relies on estimating a suitable value of the penalty parameter, which is sometimes a delicate task. Other constrained bundle-type methods, that do not use penalization, are Mifflin

(1982), Kiwiel (1985, Ch. 5). But in all of these methods serious iterates, including the starting point, are required to be feasible. This "phase I" general (nonsmooth) convex feasibility problem may be as difficult to solve as (1) itself. As a result, the overall computational burden of solving the problem may increase considerably (That said, feasible methods can be useful in applications in which problem data may not be defined everywhere outside of the feasible region).

The method considered in this work is infeasible but, if started from a feasible point, stays feasible and thus can operate "in feasible mode" if an appropriate starting point is provided. Infeasible bundle methods are very rare. We could find in the literature only the "phase I–phase II" modification of the feasible method in Kiwiel (1985, Ch. 5.7), the constrained level bundle methods of Lemaréchal et al. (1995); see also Fábián (2008), and the filter strategy of Fletcher and Leyffer (1999). However, techniques for controlling the size of subproblems had not been considered in the methods cited above; this issue had been addressed only recently, in Sagastizábal and Solodov (2005) and Karas et al. (2009).

In this paper, we use the approach of Sagastizábal and Solodov (2005). This is an infeasible proximal bundle method for solving (1), which does not use a filter or penalty function. With respect to Mifflin (1982); Kiwiel (1985), the advantage is that computing a feasible point is not needed to start the algorithm. Also, since (descent) iterates can be infeasible, monotonicity in f is not enforced (outside of the feasible set). Rather, there is a balance between the search for feasibility and for the reduction of the objective function. Furthermore, unlike the constrained level bundle method (Lemaréchal et al. 1995; Fábián 2008), we can manage the size of QP subproblems by a suitable modification of the standard bundle aggregation techniques. The approach can be viewed as an unconstrained proximal bundle method applied to the improvement function associated with problem (1). For a given $x \in \mathbb{R}^n$, this function is given by

$$h_x(y) := \max\{f(y) - f(x),\, c(y)\}, \quad y \in \mathbb{R}^n. \tag{2}$$

If the Slater regularity condition ($\exists z$ such that $c(z) < 0$) holds then, among other things, $\bar{x}$ is a solution to (1) if, and only if, $\bar{x}$ solves the unconstrained problem of minimizing $h_{\bar{x}}$ Kiwiel (1985, Lemma 2.16, p. 17). The important detail is that x in (2) is the last descent step, and thus, the function being minimized varies along the iterations. We emphasize that iterates need neither be monotone in f nor feasible. Of course, the fact that the improvement function changes along the iterations requires modifications to the method. Nevertheless, our approach is quite close to standard unconstrained bundle methods.

The rest of the paper is organized as follows. Section 2 describes the basic features of standard unconstrained bundle methods, to facilitate understanding of constrained methods, which are more complex. The section that follows describes the specific infeasible bundle algorithm for the constrained case. Section 4 presents an application to generation expansion planning, along with computational results. Finally, some concluding remarks are given.

A few words about our notation. By $\langle x, y \rangle$ we denote the inner product $\sum_{i=1}^{n} x_i y_i$ of the vectors $x, y \in \mathbb{R}^n$. Given some $\varepsilon \geq 0$, we denote the ε-subdifferential of a

convex function h at the point $x \in \mathbb{R}^n$ by $\partial_\varepsilon h(x) = \{g \in \mathbb{R}^n \mid h(y) \geq h(x) + \langle g, y - x \rangle - \varepsilon \, \forall \, y \in \mathbb{R}^n\}$, with $\partial_0 h(x) = \partial h(x)$ being the usual subdifferential.

2 An overview of bundle methods

Since bundle methods are relatively complex, we start by introducing the main ideas for the case of the standard unconstrained problem

$$\min_{x \in \mathbb{R}^n} h(x),$$

where $h(\cdot)$ is some fixed convex function. For the sake of simplicity, we also do not discuss for now the ways to control the size of subproblems (i.e., no bundle compression/aggregation).

Let ℓ be the current iteration index. Bundle methods keep memory of the past in a bundle of information

$$\mathcal{B}_\ell := \bigcup_{i < \ell} \left\{ \left(y^i, h_i = h(y^i), g_h^i \in \partial h(y^i) \right) \right\} \text{ and } (x^k, h(x^k)), \ k = k(\ell),$$

where $k(\ell)$ denotes the index of the last descent step (also called serious step) preceding the iteration ℓ. Serious iterates form a subsequence $\{x^k\} \subset \{y^i\}$ such that $\{h(x^k)\}$ is strictly decreasing.

The bundle of past information is used to define at each iteration a cutting-plane model of the objective function

$$\psi_\ell(y) := \max_{i \in \mathcal{B}_\ell} \left\{ h_i + \langle g_h^i, y - y^i \rangle \right\}. \tag{3}$$

An equivalent expression, better suited for implementations, centers the cutting-plane model at x^k:

$$\psi_\ell(y) = h(x^k) + \max_{i \in \mathcal{B}_\ell} \left\{ -e_i^k + \langle g_h^i, y - x^k \rangle \right\}, \tag{4}$$

where the terms e_i^k are the (nonnegative) linearization errors

$$e_i^k := h(x^k) - h_i - \langle g_h^i, x^k - y^i \rangle.$$

In particular,

$$g_h^i \in \partial_{e_i^k} h(x^k).$$

Since the linearization errors depend on x^k, they need to be updated every time x^k changes. For further reference, note that the linearization errors obviously "depend" also on h (h is fixed in this section, but not in the rest of the paper). Thus in the

update of the linearization errors in our algorithm, we shall also have to account for an eventual change of the objective function h.

The advantage of expressing the cutting-plane model (3) in the form of (4) is that the later requires less memory for storing the relevant information than the former: the bundle becomes

$$\mathcal{B}_\ell = \bigcup_{i < \ell} \left\{ \left(e_i^k \in \mathbb{R}_+, g_h^i \in \partial_{e_i^k} h(x^k) \right) \right\} \text{ and } (x^k, h(x^k)).$$

Given proximal (stabilization) parameter $\mu_\ell > 0$, the next iterate y^ℓ is generated by solving the QP reformulation of the problem

$$\min_{y \in \mathbb{R}^n} \psi_\ell(y) + \frac{1}{2} \mu_\ell \|y - x^k\|^2. \tag{5}$$

Clearly, y^ℓ is unique (by strong convexity of the objective function). Furthermore, it is characterized by the following conditions [see Bonnans et al. (2006, Lemma 9.8)]:

$$y^\ell = x^k - \frac{1}{\mu_\ell} \hat{g}^\ell, \text{ where } \hat{g}^\ell \in \partial \psi_\ell(y^\ell),$$

$$\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h(x^k), \text{ where } \hat{\varepsilon}_\ell^k = h(x^k) - \psi_\ell(y^\ell) - \frac{1}{\mu_\ell} \|\hat{g}^\ell\|^2 \geq 0.$$

An iterate y^ℓ is considered good enough to become the next serious step when $h(y^\ell)$ provides a significant decrease relative the decrease predicted by the model. Specifically, let $m \in (0, 1)$ be a given parameter. The decrease predicted by the model is given by

$$\delta_\ell := h(x^k) - \psi_\ell(y^\ell) - \frac{1}{2} \mu_\ell \|y^\ell - x^k\|^2 = \hat{\varepsilon}_\ell^k + \frac{1}{2\mu_\ell} \|\hat{g}^\ell\|^2 \geq 0.$$

When y^ℓ satisfies the descent test

$$h(y^\ell) \leq h(x^k) - m\delta_\ell, \tag{6}$$

a serious step is declared: $x^{k+1} = y^\ell$. Otherwise, y^ℓ is declared a null step and x^k remains unchanged.

2.1 Aggregation technique

The number of constraints in the QP reformulation of the problem (5) used to generate y^ℓ is precisely the number of elements in the bundle $\mathcal{B}_\ell$. Obviously, one has to keep this number computationally manageable. Thus, the bundle has to be compressed when the number of elements reaches some software-imposed bound. This has to be

done without impairing convergence of the algorithm. For this purpose, the so-called aggregate function is fundamental:

$$l_{k,\ell}(y) := h(x^k) - \hat{\varepsilon}_\ell^k + \langle \hat{g}^\ell, y - x^k \rangle, \quad k = k(\ell).$$

Before looping from ℓ to $\ell + 1$, the next bundle $\mathcal{B}_{\ell+1}$ is defined. If the bundle has reached its maximal allowed size, it must be "compressed". Compressing the bundle amounts to replacing (at iteration $\ell + 1$) the cutting-plane model (4) by another function, defined with a smaller number of cutting planes, which we shall still denote by $\psi_{\ell+1}$. As pointed out in Correa and Lemaréchal (1993, Sect. 4, Eqs. (4.7)–(4.9)), one can use any collection of functions satisfying (for all $y \in \mathbb{R}^n$) the following three conditions:

$$\psi_\ell(y) \leq h(y) \qquad \text{for all } \ell \geq 1, \qquad (7a)$$

$$l_{k(\ell),\ell}(y) \leq \psi_{\ell+1}(y) \qquad \text{for those } \ell \text{ for which } y^\ell \text{ is a null step}, \qquad (7b)$$

$$h_\ell + \langle g_h^\ell, y - y^\ell \rangle \leq \psi_{\ell+1}(y) \qquad \text{for those } \ell \text{ for which } y^\ell \text{ is a null step}. \qquad (7c)$$

We note that (7a) will not be automatic in our setting. Indeed, as already commented, the function h will be changing after every serious step. As a consequence, satisfaction of (7a) is not straightforward, and appropriate care needs to be taken.

Suppose, however, that (7a) holds. In terms of bundle information, the remaining conditions mean that it is enough for the new bundle to contain both the aggregate information (to ensure (7b)) and the last generated information (to ensure (7c)). These values are, respectively, $(\hat{\varepsilon}_\ell^k, \hat{g}^\ell)$ and $(y^\ell, h_\ell, g_h^\ell \in \partial h(y^\ell))$. In particular, at any iteration, the bundle can contain as few elements as we wish (as long as the two elements specified above are included).

Accordingly, we shall write the next bundle in the form

$$\mathcal{B}_{\ell+1} := \mathcal{B}_{\ell+1}^{oracle} \bigcup \mathcal{B}_{\ell+1}^{agg} \quad \text{and } (x^k, h(x^k)), \ k = k(\ell + 1), \text{ the last serious iterate},$$

where the "oracle" bundle is any set such that

$$\left\{ \left(e_\ell^k, g_h^\ell \right) \right\} \subseteq \mathcal{B}_{\ell+1}^{oracle} \subseteq \bigcup_{i \leq \ell} \left\{ \left(e_i^k \in \mathbb{R}_+, g_h^i \in \partial_{e_i^k} h(x^k) \right) \right\},$$

while the "aggregate" bundle satisfies

$$\left\{ \left(\hat{\varepsilon}_\ell^k, \hat{g}^\ell \right) \right\} \subseteq \mathcal{B}_{\ell+1}^{agg} \subseteq \bigcup_{i \leq \ell} \left\{ \left(\hat{\varepsilon}_i^k \in \mathbb{R}_+, \hat{g}^i \in \partial_{\hat{\varepsilon}_i^k} h(x^k) \right) \right\}.$$

The left-most inclusions in the last two relations above need to be specified explicitly only when there is bundle compression at the ℓ-th iteration (if there is no compression, they hold automatically, because of the right-most inclusions). We note that similarly

to updating the linearization errors e_i^k, the quantities $\hat{\varepsilon}_i^k$ also need to be updated every time when k changes, see (8) and (14) below.

The next cutting-plane model is then defined by

$$\psi_{\ell+1}(y)$$
$$= h(x^k) + \max\left\{ \max_{i \in \mathcal{B}_{\ell+1}^{oracle}} \left\{ -e_i^k + \langle g_h^i, y - x^k \rangle \right\}, \max_{i \in \mathcal{B}_{\ell+1}^{agg}} \left\{ -\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k \rangle \right\} \right\},$$
$$k = k(\ell + 1).$$

As already mentioned, every time a new serious step has been declared, both linearization and aggregate errors need to be modified. The update aims at satisfying the key relations

$$g_h^i \in \partial_{e_i^k} h(x^{k+1}) \quad \text{and} \quad \hat{g}^i \in \partial_{\hat{\varepsilon}_i^k} h(x^{k+1}),$$

which should hold for all elements in the new bundle. The following simple updating formulæ guarantee the required properties (when h is fixed):

$$\begin{cases} e_i^{k+1} := e_i^k + h(x^{k+1}) - h(x^k) + \langle g_h^i, x^k - x^{k+1} \rangle & \text{if } i \in \mathcal{B}_{\ell+1}^{oracle}, \\ \hat{\varepsilon}_i^{k+1} := \hat{\varepsilon}_i^k + h(x^{k+1}) - h(x^k) + \langle \hat{g}^i, x^k - x^{k+1} \rangle & \text{if } i \in \mathcal{B}_{\ell+1}^{agg}. \end{cases} \quad (8)$$

3 A constrained bundle algorithm

Given the last serious iterate x^k, we apply an unconstrained proximal bundle method to the function $h(\cdot) := h_k(\cdot) = h_{x^k}(\cdot)$, defined by (2), until the next serious iterate x^{k+1} is generated. At this time, we change $h(\cdot)$ to $h_{k+1}(\cdot) = h_{x^{k+1}}(\cdot)$, make the necessary modifications to the bundle, and repeat the process. We point out that the development is not straightforward. For one thing, it is possible that $f(x^{k+1}) > f(x^k)$. As is easy to observe, in that case we have $h_{k+1}(\cdot) \le h_k(\cdot)$. As a consequence, the accumulated cutting-plane model for $h_k(\cdot)$ may not be a valid (lower) approximation for $h_{k+1}(\cdot)$. Thus, the model has to be revised and adjusted to ensure that conditions (7a)–(7c) (in particular (7a)) are satisfied for the new $h(\cdot) := h_{k+1}(\cdot)$. Note that this adjustment is independent of compressing the bundle, which will require additional care.

3.1 Bundle information

Since $h(\cdot)$ varies with k, past information relevant for constructing the model is no longer just (e_i, g_h^i). In particular, separate information about the objective and constraint functions needs to be kept. This information is $(f_i = f(y^i), c_i = c(y^i))$ and $(g_f^i \in \partial f(y^i), g_c^i \in \partial c(y^i))$. Or, equivalently, $(e_{f_i}^k, e_{c_i}^k, g_f^i \in \partial_{e_{f_i}^k} f(x^k), g_c^i \in \partial_{e_{c_i}^k} c(x^k))$, where the linearization errors for f and c, respectively, are

$$e_{f_i}^k := f(x^k) - f_i - \langle g_f^i, x^k - y^i \rangle, \quad e_{c_i}^k := c(x^k) - c_i - \langle g_c^i, x^k - y^i \rangle. \quad (9)$$

The purpose of keeping the bundle information separated is twofold:

– First, knowing (f_i, c_i) makes it possible to compute the function and subgradient values for different functions h.
– Second, separate linearization errors can be updated by a simple formula, even when h changes.

Therefore, we define

$$\mathcal{B}_\ell := \mathcal{B}_\ell^{oracle} \bigcup \mathcal{B}_\ell^{agg} \quad \text{and} \quad (x^k, f(x^k), c(x^k)), \ k = k(\ell), \ \text{the last serious iterate,}$$
$$\mathcal{B}_\ell^{oracle} \subseteq \bigcup_{i<\ell} \left\{ \left(f_i, c_i, e_{f_i}^k, e_{c_i}^k, g_f^i \in \partial_{e_{f_i}^k} f(x^k), g_c^i \in \partial_{e_{c_i}^k} c(x^k) \right) \right\},$$
$$\mathcal{B}_\ell^{agg} \subseteq \bigcup_{i<\ell} \left\{ \left(\hat{\varepsilon}_i^k, \hat{g}^i \in \partial_{\hat{\varepsilon}_i^k} h_k(x^k) \right) \right\}. \tag{10}$$

Let $c^+(x) = \max\{c(x), 0\}$. In the notation of (9) and (10), for each $i \in \mathcal{B}_\ell^{oracle}$, define

$$\begin{cases} e_i^k := e_{f_i}^k + c^+(x^k) & \text{and } g_{h_k}^i := g_f^i, \text{ if } f_i - f(x^k) \geq c_i, \\ e_i^k := e_{c_i}^k + c^+(x^k) - c(x^k) & \text{and } g_{h_k}^i := g_c^i, \text{ if } f_i - f(x^k) < c_i. \end{cases} \tag{11}$$

Then $e_i^k \geq 0$ and $g_{h_k}^i \in \partial_{e_i^k} h_k(x^k)$; see Sagastizábal and Solodov (2005) for the proof.

The cutting-plane model associated with (10), (11) is given by

$$\psi_\ell(y) = c^+(x^k) + \max\left\{ \max_{i \in \mathcal{B}_\ell^{oracle}} \left\{ -e_i^k + \langle g_{h_k}^i, y - x^k \rangle \right\}, \right.$$
$$\left. \max_{i \in \mathcal{B}_\ell^{agg}} \left\{ -\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k \rangle \right\} \right\}, \qquad k = k(\ell). \tag{12}$$

For this model to satisfy (7a)–(7c) when passing to the iteration $\ell + 1$, we consider separately the two cases of the ℓ-th iteration being a null step or a serious step.

Suppose first that the QP subproblem defined with ψ_ℓ given by (12) generates y^ℓ as a null step. By construction, the new bundle satisfies (10) and (11) written with ℓ replaced by $\ell + 1$ (k remains the same). Thus, $g_{h_k}^i \in \partial_{e_i^k} h_k(x^k)$ for all $i \in \mathcal{B}_{\ell+1}^{oracle}$. Likewise, aggregate subgradients satisfy the inclusion $\hat{g}^i \in \partial_{\hat{\varepsilon}_i^k} h_k(x^k)$ for all $i \in \mathcal{B}_{\ell+1}^{agg}$. Therefore, (7a) (written with ℓ replaced by $\ell + 1$) is automatically satisfied. Finally, for conditions (7b) and (7c) to hold, it is enough to make sure that

$$\left\{ \left(e_\ell^k, g_{h_k}^\ell \in \partial_{e_\ell^k} h_k(x^k) \right) \right\} \subseteq \mathcal{B}_{\ell+1}^{oracle} \text{ and } \left\{ \left(\hat{\varepsilon}_\ell^k, \hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h_k(x^k) \right) \right\}$$
$$\subseteq \mathcal{B}_{\ell+1}^{agg}, \text{ if there is compression.}$$

Those inclusions are also automatically satisfied, if the bundle is managed as in any standard method; see Step 4 in Algorithm 3.1 below.

Therefore, when there is a null step, the update of the bundle (and of the model) does not present any problem. This is as expected, since the function $h(\cdot) = h_k(\cdot)$ is fixed between consecutive serious steps. The situation changes when y^ℓ is declared a serious step. Specifically, the aggregate bundle elements need a special update.

3.2 Adjusting the model

Suppose that for some iteration ℓ the descent test is satisfied (i.e., condition (6) written with h replaced by h_k), so that a new stability center $x^{k+1} = y^\ell$ is generated. This means, in particular, that at the next iterate we start working with the new function $h_{k+1}(\cdot) = h_{x^{k+1}}(\cdot)$.

As mentioned in Correa and Lemaréchal (1993), conditions (7a)–(7c) guarantee that the bundle technique applied to the new function $h(\cdot) = h_{k+1}(\cdot)$ either produces a descent step after a finite number of null steps, or the point x^{k+1} is a minimum of $h_{k+1}(\cdot)$. However, condition (7a) (written with $\ell = \ell + 1$) is not automatic in our setting, and the model may need to be properly adjusted. Indeed, even though

$$\psi_\ell(y) \leq h_k(y) \text{ and } c^+(x^k) + \langle \hat{g}^i, y - x^k \rangle - \hat{\varepsilon}_i^k \leq h_k(y), \ i \in \mathcal{B}_\ell^{agg},$$

the same inequalities may not hold with h_k replaced by h_{k+1}. Specifically, if $f(x^k) < f(x^{k+1})$, which is possible, then we have that $h_k(y) \geq h_{k+1}(y)$. Thus, the key relations (7a)–(7c) are not guaranteed and in general do not hold.

There are various ways to ensure (7a)–(7c) after a serious step is taken. We next present one approach.

Let ψ_ℓ be defined by (12), using (10) and (11). Suppose that the associated y^ℓ is declared a serious step, i.e., $x^{k+1} = y^\ell$. Then the following holds:

(i)   For each $i \in \mathcal{B}_\ell^{oracle}$, the linearization errors

$$\begin{aligned} e_{f_i}^{k+1} &= e_{f_i}^k + f(x^{k+1}) - f(x^k) + \langle g_f^i, x^k - x^{k+1} \rangle, \\ e_{c_i}^{k+1} &= e_{c_i}^k + c(x^{k+1}) - c(x^k) + \langle g_c^i, x^k - x^{k+1} \rangle \end{aligned} \tag{13}$$

satisfy (9) written with $k = k + 1$. As a result, $g_{h_{k+1}}^i \in \partial_{e_i^{k+1}} h_{k+1}(x^{k+1})$, where $e_i^{k+1} \geq 0$ and $g_{h_{k+1}}^i$ are defined in (11), written with k replaced by $k + 1$.

(ii) For each $i \in \mathcal{B}_\ell^{agg}$, define

$$\hat{\varepsilon}_i^{k+1} := \hat{\varepsilon}_i^k + c^+(x^{k+1}) - c^+(x^k) + \left(f(x^{k+1}) - f(x^k) \right)^+ + \langle \hat{g}^i, x^k - x^{k+1} \rangle. \tag{14}$$

Then $\hat{\varepsilon}_i^{k+1} \geq 0$ and $\hat{g}^i \in \partial_{\hat{\varepsilon}_i^{k+1}} h_{k+1}(x^{k+1})$; see Sagastizábal and Solodov (2005) for a proof.

As a consequence, regardless of whether the ℓ-th iteration produced a null step or a serious step, if

$$\mathcal{B}_{\ell+1}^{oracle} \subseteq \bigcup_{i \leq \ell} \left\{ \left(f_i, c_i, e_{f_i}^{k+1}, e_{c_i}^{k+1}, g_f^i, g_c^i \right) \right\} \text{ and } \mathcal{B}_{\ell+1}^{agg} \subseteq \bigcup_{i \leq \ell} \left\{ \left(\hat{\varepsilon}_i^{k+1}, \hat{g}^i \right) \right\},$$

then the model

$$\psi_{\ell+1}(y) = c^+(x^k) + \max \left\{ \max_{i \in \mathcal{B}_{\ell+1}^{oracle}} \left\{ -e_i^k + \langle g_{h_k}^i, y - x^k \rangle \right\}, \right.$$

$$\left. \max_{i \in \mathcal{B}_{\ell+1}^{agg}} \left\{ -\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k \rangle \right\} \right\},$$

$k = k(\ell + 1)$, satisfies (7a) written with ℓ replaced by $\ell + 1$, with $h(\cdot) = h_k(\cdot)$, $k = k(\ell + 1)$. Furthermore, if $\left(\hat{\varepsilon}_\ell^k, \hat{g}^\ell \right) \subseteq \mathcal{B}_{\ell+1}^{agg}$, then $\psi_{\ell+1}$ satisfies (7b), and if $\left(f_\ell, c_\ell, e_{f_\ell}, e_{c_\ell}, g_f^\ell, g_c^\ell \right) \subseteq \mathcal{B}_{\ell+1}^{oracle}$, then $\psi_{\ell+1}$ satisfies (7c).

3.3 An infeasible constrained bundle method

We are now in a position to give the algorithm in full detail.

Algorithm 3.1 (Infeasible Constrained Proximal Bundle Method (ICPBM))

Step 0. Initialization.
   Choose parameters $m \in (0, 1)$, $tol \geq 0$, and an integer $|\mathcal{B}|_{max} \geq 2$. Choose $x^0 \in \mathbb{R}^n$. Set $y^0 := x^0$, and compute (f_0, c_0, g_f^0, g_c^0). Set $k = 0$, $\ell = 1$, $e_{f_0} := 0$, $e_{c_0} := 0$ and define the starting bundles $\mathcal{B}_1^{oracle} := \{(e_{f_0}^0, e_{c_0}^0, f_0, c_0, g_f^0, g_c^0)\}$ and $\mathcal{B}_1^{agg} := \emptyset$.

Step 1. Quadratic Programming Subproblem.
   Choose $\mu_\ell > 0$ and compute y^ℓ as the solution to

$$\min_{y \in \mathbb{R}^n} \psi_\ell(y) + \frac{1}{2}\mu_\ell \|y - x^k\|^2, \tag{15}$$

   where ψ_ℓ is defined by (12) and (11). Compute

$$\hat{g}^\ell = \mu_\ell(x^k - y^\ell), \quad \hat{\varepsilon}_\ell^k = c^+(x^k) - \psi_\ell(y^\ell) - \frac{1}{\mu_\ell}\|\hat{g}^\ell\|^2,$$

$$\delta_\ell = \hat{\varepsilon}_\ell^k + \frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2.$$

   Compute $(f_\ell, c_\ell, g_f^\ell, g_c^\ell)$ and $(e_{f_\ell}^k, e_{c_\ell}^k)$, using (9) written with $i = \ell$.

Step 2. Stopping test. *If $\delta_\ell \leq tol$, stop.*

Step 3. Descent test. *Compute $h_\ell := h_k(y^\ell) = \max\{f_\ell - f(x^k), c_\ell\}$. If $h_\ell \leq c^+(x^k) - m\delta_\ell$, then declare a serious step. Otherwise, declare a null step.*

Step 4. `Bundle Management.`
   Set $\mathcal{B}_{\ell+1}^{oracle} := \mathcal{B}_{\ell}^{oracle}$ and $\mathcal{B}_{\ell+1}^{agg} := \mathcal{B}_{\ell}^{agg}$.
   If the bundle has reached the maximum bundle size, i.e., if $|\mathcal{B}_{\ell+1}^{oracle} \cup \mathcal{B}_{\ell+1}^{agg}| = |\mathcal{B}|_{max}$, then:
   Delete at least two elements from $\mathcal{B}_{\ell+1}^{oracle} \cup \mathcal{B}_{\ell+1}^{agg}$, insert the aggregate couple $(\hat{\varepsilon}_{\ell}^{k}, \hat{g}^{\ell})$ in $\mathcal{B}_{\ell+1}^{agg}$, append $(e_{f_{\ell}}^{k}, e_{c_{\ell}}^{k}, f_{\ell}, c_{\ell}, g_{f}^{\ell}, g_{c}^{\ell})$ to $\mathcal{B}_{\ell+1}^{oracle}$.

Step 5. `Model adjustment (serious step).`
   If y^{ℓ} is a serious step, then: Define the next stability center $(x^{k+1}, f(x^{k+1}), c(x^{k+1})) := (y^{\ell}, f_{\ell}, c_{\ell})$, update the linearization errors for $i \in \mathcal{B}_{\ell+1}^{oracle}$ using (13), update the aggregate errors for $i \in \mathcal{B}_{\ell+1}^{agg}$ using (14). Set $k = k+1$.

Loop. *Set $\ell = \ell + 1$ and go to Step 1.*

Some remarks are in order. Recalling the definition of $h_k(\cdot)$, we conclude that if the descent test is satisfied and a serious step is declared, then it must hold that

$$f(x^{k+1}) - f(x^k) \leq c^+(x^k) - m\delta_{\ell} \quad \text{and} \quad c(x^{k+1}) \leq c^+(x^k) - m\delta_{\ell}.$$

In particular, if x^k is infeasible, then $f(x^{k+1}) > f(x^k)$ is possible (since $c^+(x^k) > 0$). Therefore, the method is not monotone with respect to f when outside of the feasible region. However, outside of the feasible region it is monotone with respect to c, because $c(x^{k+1}) < c^+(x^k) = c(x^k)$ for x^k infeasible. This seems intuitively reasonable: while it is natural to accept the increase in the objective function value in order to decrease infeasibility, it is not so clear why one would want to decrease the objective function at the expense of moving away from the feasible region. The situation reverses when x^k is feasible. In that case, $c^+(x^k) = 0$, so that $f(x^{k+1}) < f(x^k)$. But although the descent test above implies that x^{k+1} is feasible too, it is possible that $c(x^{k+1}) > c(x^k)$ (except when $c(x^k)$ is exactly zero). This also appears completely reasonable: while preserving feasibility, we allow c to increase (so that the boundary of the feasible set can be approached), at the same time obtaining a decrease in the objective function.

3.4 Convergence properties

We summarize convergence properties without going into details, which can be found in Sagastizábal and Solodov (2005). Suppose (1) satisfies the Slater constraint qualification, and its solution set is nonempty.

   If Algorithm 3.1 generates an infinite sequence of serious steps which is bounded (this holds, for example, if any iterate is feasible), then the sequence $\{x^k\}$ converges to a solution to (1).

   If Algorithm 3.1 makes a finite number of serious steps (i.e., from some point on all the iterations are null steps), then the last serious iterate is a solution to (1).

4 Generation expansion problems

We next describe a variant of the generation expansion planning problem (GEP) in Marcato and Sagastizábal (2007), and its solution by the infeasible constrained bundle algorithm.

4.1 Test problem

We consider GEP along the lines of Marcato and Sagastizábal (2007), with one important difference. Namely, besides air pollution, we take into account the fact that some thermal power plants also pollute rivers and lakes when the remains of burned fuel are removed from the burner using water. The additional consideration of water pollution introduces a convex constraint in GEP, cf. (16) below, requiring eventually to solve a constrained nonsmooth problem.

The model considers different scenarios for air pollution limits and for the future demand. We do not place the problem in a competitive economic environment, but rather in a vertically integrated system. We consider a portfolio of investments composed by candidate projects of two types: projects adding capacity to existing plants (with a concave piecewise linear cost curve for thermal plants) and projects for constructing new plants, which can be compulsory, mutually exclusive, or complementary. New coal plants can be standard, or using the so-called flue-gas desulfurization (fgd) technique to reduce air pollution. Coal of both types (high and low sulfur), can be bought in limited quantities on the spot market in each period. In addition, long term contracts are available for ensuring the supply of low sulfur coal in large quantities along the planning horizon time of N periods. However, these (take-or-pay) contracts involve the commitment of a minimum purchase in each period. Generation costs include operating and fuel costs. Load constraints are given by blocks (base, intermediate, peak), with a reserve margin in the peak block. We refer to Marcato and Sagastizábal (2007) for full details of the model. We recall here its main features.

Uncertainties.   For a planning horizon of N periods, uncertainties regarding future demand and air pollution limits are modeled in a scenario tree as in Fig. 1, which shows that all uncertainties are revealed at the end of the N_1^{th}-period. From this period on, the optimization process selects optimal decisions for each one of the Q scenarios in the last $N - N_1$ periods of the planning horizon.

Each branch q in the scenario tree has a probability denoted by p_q, corresponding to the joint probability of the scenario uncertainties. For example, if scenario $q = 1$ corresponds to the case of low economic growth and stringent pollution control, each one with probability 0.3, then $p_1 = 0.09$.
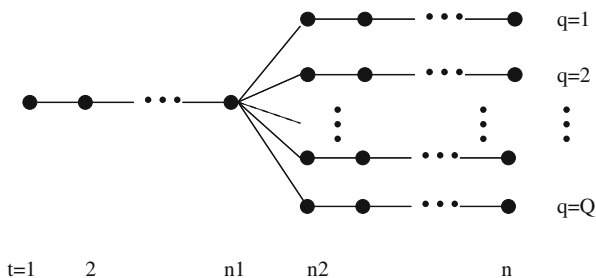
Fig. 1 Scenario tree representing uncertainties

Variables. Physical variables of the problem are: added capacity, low sulfur coal bought with a given contract, coal bought in the spot market (low or high sulfur coal), and generation $gen_{v_i}^t$ of a given project i at a given time period t, and for $v =$ base, peak, and intermediate demand block. There are also modeling binary variables, for the different contracts and also for the different investment projects, denoted by u_i^t. Depending on the specific time period they correspond to, variables will be separated in first and second stage. For notational convenience, all variables in periods 1 to N_1 are gathered in the first stage vector x. Second stage variables correspond to periods $N_1 + 1$ to N, and are denoted by $\tilde{x}$, with a subindex q to distinguish different scenarios, if needed for clarity.

Coal plants. Thermal power plants consume coal or natural gas and pollute the air, and, in some cases, nearby water resources. Combustion of coal or natural gas increases the level of CO_2 and SO_2 in the atmosphere. For coal-fired plants the demand constraint incorporating the percentage of SO_2 by weight through a coefficient reflects a nature of the sulfur content of each ton of coal.

Constraints. As mentioned, load constraints are given by blocks (base, intermediate, peak), with a reserve margin in the peak block.

Air pollution constraints are expressed in the form of maximum limits of the total sulfur emission allowed for the coal mix for each environmental scenario. Clearly, the emission produced when burning low sulfur coal is less than when burning high sulfur coal. Moreover, for fgd thermal plants both levels of emission are lower. These environmental constraints are affine and can be expressed in an abstract form as $x \in X_{\text{aff}}$, or $\tilde{x} \in \tilde{X}_{\text{aff}}$, depending on the stage.

Water pollution constraints result from cleansing procedures for thermal plants, which can be done at fixed time steps, whenever the remaining fuel falls below a lower limit known a priori, say $\texttt{clean}_i$ for the ith thermal plant. Following Avetisyan et al. (2006), this type of pollution can be modeled by limiting the concentration of pollutants in water resources. In turn, such concentration is represented as a decreasing convex function of how much fuel remains: the less combustible remains, the more pollutant residuals lie in the bottom of the tank. The amount of remaining fuel varies in an inverse manner with the amount of fuel burnt so far, which in turn depends on the generation until time period t: $\sum_{\tau=1}^t \sum_{v=1}^3 gen_{v_i}^\tau$. We use a scaling parameter θ_i to convert energy generation into burnt fuel and let $\texttt{fuel}_i^0$ be the initial amount of combustible at plant i. Accordingly, the fuel available until the cleansing process is triggered is

$$\texttt{rem}_i^t = \max\left(0, \texttt{fuel}_i^0 - \theta_i \sum_{\tau=1}^t \sum_{v=1}^3 gen_{v_i}^\tau - \texttt{clean}_i\right),$$

and the water pollution constraint takes the form:

$$e^{-\alpha_i \texttt{rem}_i^t} \leq 1 + con_{max}, \text{ a convex constraint with respect to } \texttt{rem}_i^t;$$
$$\text{hence also to } x. \tag{16}$$

In the expression above, con_{max} stands for the maximum allowed concentration of pollutant, while α is the associated reaction rate coefficient.

Thermal pollution, i.e., increases in water temperature caused when using water as a coolant, can be modeled in a similar manner.

Air pollution constraints do not couple different time periods. By contrast, some investment constraints couple first and second stage variables. Suppose some project i_c is compulsory, i.e., it has to be built during the planning horizon. For each scenario $q \leq Q$, the modeling binary variables $u_{i_c}^t$ for $t = 1, \ldots, 4$, and $u_{q_{i_c}}^t$ for $t = 5, \ldots, 10$, must satisfy the relation coupling first and second stage variables:

$$\sum_{t=1}^{N_1} u1_{i_c}^t + \sum_{t=N_1+1}^{N} u_{q_{i_c}}^t = 1, \quad \text{an affine constraint with respect to } x \text{ and } \tilde{x}. \quad (17)$$

Non-compulsory projects introduce similar coupling, but for inequality constraints.

The resulting model is a two-stage stochastic programming problem with recourse of the form:

$$\begin{cases} \min \langle f_o, x \rangle + \langle \tilde{f}_o, \tilde{x} \rangle \\ x \in X := \text{lin}(X_{\text{box}}) \cap X_{\text{aff}} \\ \tilde{x} \in \tilde{X} := \text{lin}(\tilde{X}_{\text{box}}) \cap \tilde{X}_{\text{aff}} \\ Ax + \tilde{A}\tilde{x} \leq b \\ c(x) \leq 0. \end{cases} \quad (18)$$

The objective function (operating, fuel, cleansing, and investment costs) is linear and separable by stages and scenarios. In the feasible set of (18) the notation $\text{lin}(X_{\text{box}})$ $(\text{lin}(\tilde{X}_{\text{box}}))$ is used to emphasize that each 0-1 variable was replaced by a continuous variable ranging in the interval [0, 1]. Water pollution constraints, denoted abstractly by $c(x)$ in (18), are considered in the nonlinear form (16) only for first stage time periods. For second stage periods, a linearized formulation is used, and, hence, it enters the set $\tilde{X}_{\text{aff}}$. As suggested by a reviewer, instead of linearizing the water pollution limits for the second stage, a worst-case approach convex risk-measure (over the random variable of generation) could be used. This is an interesting subject for future research.

Finally, the coupling constraint $Ax + \tilde{A}\tilde{x} \leq b$ represents in a compact form coupling constraints such as (17), which couple first and second order stages, but not scenarios. We make the approximations above of the original mixed-integer nonlinear programming problem in order to ensure convergence of the decomposition method applied for solving (18) with probability one.

4.2 Decomposition method

Problem (18) has a structure suitable for the application of a Benders'-like decomposition method. Specifically, consider the following problem, equivalent to (18):

$$\begin{cases} \min \ \langle f_o, x \rangle + v^{opt}(x) \\ x \in X \\ c(x) \le 0, \end{cases} \tag{19}$$

where we defined the *optimality function*

$$v^{opt}(x) = \begin{cases} \min \ \langle \tilde{f}_o, \tilde{x} \rangle \\ \tilde{x} \in \tilde{X} \\ \tilde{A}\tilde{x} \le b - Ax. \end{cases} \tag{20}$$

For our relaxed problem (18), where all the data is convex and the cost function has additive form, v^{opt} is a convex nondifferentiable function with separable structure, by scenarios. For a given first stage vector x, the vector $A^\top \pi^*(x)$, with $\pi^*(x)$ being an optimal multiplier corresponding to the inequality constraint in (20), is a subgradient for v^{opt} at x.

The reformulation of (18) using two levels yields a decomposition method, composed by a *master* program (19) and a *slave* problem (20), decomposable in Q linear programs, one per scenario. The master program is a constrained nonsmooth optimization problem of the form (1).

For simplicity, we assume that problem (18) has *relatively complete recourse*, i.e., for all $x \in X$ the feasible set in (20) is nonempty for all scenarios, with probability one. For GEP this property is ensured by introducing a fictitious plant with large enough capacity and operating cost equal to the deficit cost resulting from load shedding.

4.3 Results

For illustration, we apply algorithm ICPBM to a prototype system composed by nuclear, gas, oil, and coal plants. We consider a mix from Bienstock and Shapiro (1988), composed by 11 existing plants using coal (3 plants), coal with fgd (2), oil (2), peakers (3), and nuclear (1). A peaking power plant, or "peaker", generally runs only when there is a high demand, known as peak demand, for electricity.

There are 5 capacity expansion options (for coal, coal with fgd, oil, gas, 1 and peaker plants). The full set of data can be found in Bienstock and Shapiro (1988).

We present results for different demand scenarios (low, medium and high economic growth) and environmental cases (non-stringent and stringent), according to the data in Table 1.

Table 1 Scenarios for the test case

Scenarios	Demand growth		
	1% ($prob.$ 0.3)	2% ($prob.$ 0.4)	4% ($prob.$ 0.3)
Maximum air pollution allowed			
0.4% ($prob.$ 0.3)	1	3	5
0.7% ($prob.$ 0.7)	2	4	6

Note that scenario 5 is the one with strongest constraints (highest demand growth and lowest pollution limit), while scenario 2 is the less stringent one. Scenarios 2, 4, and 6 have less stringent control on sulfur emissions. Scenarios 1 and 2 (respectively, 5 and 6) have the lowest (respectively, the highest) demand.

The values taken for the water pollution constraint (16) were the following. For all plants and first stage periods, rate $\alpha = 0.15$, and $con_{max} = 0.05$. For each coal plant, the value of $clean_i$ triggering the cleansing procedure was set to 0.5% of the capacity of the plant, and the cleansing cost was 10% of the generation cost.

The initial mix configuration was taken as starting point, and the stopping tolerance was set to $5 * 10^{-5}$. We made runs on two variants of the test case. The first variant, referred to as "without WP control", does not consider the water pollution (WP) constraint (16). The second variant considers (16) and is referred accordingly as "with WP control". For the variant without WP the method converged to the optimal value in 12 iterations. For the variant with WP control the method converged in 14 iterations. The final constraint violation was less than 10^{-4} in all cases.

Our main results are reported in Figs. 2, 3, 4, 5, 6 below. Each figure contains two graphs, the left one corresponding to the values obtained without WP control, while the right one corresponds to the variant with WP control. The reported results compare the consumption of low sulfur (LS) and high sulfur (HS) coal for the planning period, the expansion plan, and the sources of generation (no coal, fgd-coal, polluting coal plants) for the first stage period, as well as for the two extreme scenarios, namely #2 and #5.

In Fig. 2 we see that, with or without WP control, LS coal levels are kept constant for all periods and scenarios. However, as a result of constraint (16), we observe a drop in such levels, which fall from 1.4 millions of BTU without WP control to 1.2 millions of BTU if there is WP control (BTU stands for British Thermal Unit, a traditional unit of energy most often used in the power industry).

As for the sources of LS coal, with WP control there is practically no LS coal bought in the spot market, as opposed to the variant without WP control. The amount of contracted coal is similar in both variants.
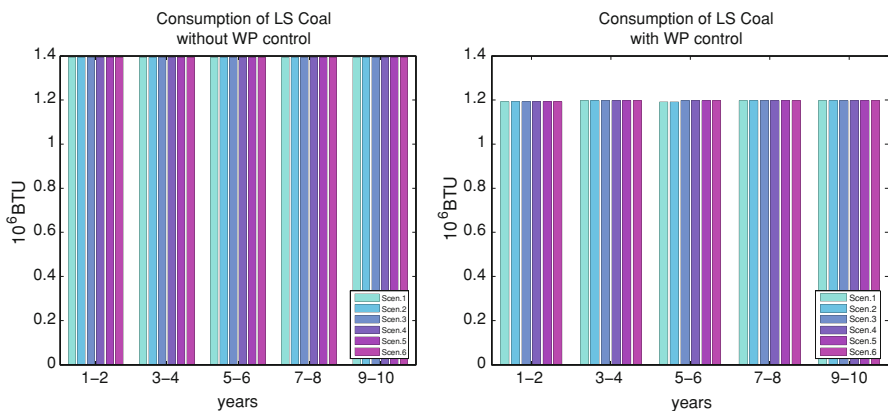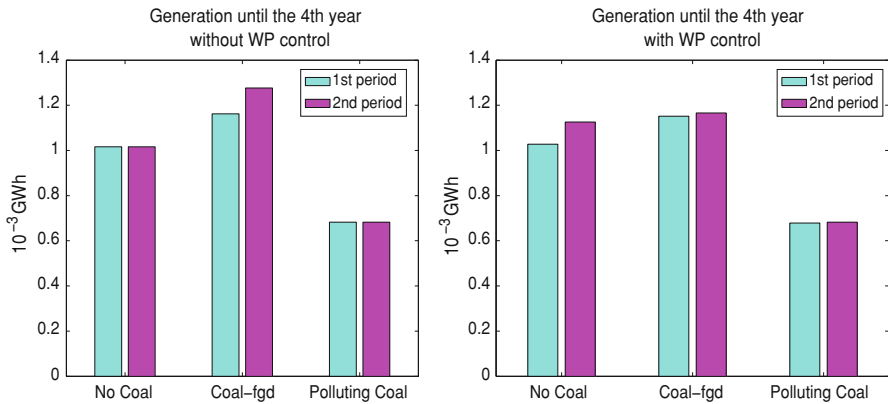

Fig. 2 LS coal consumption, without WP control (*left*) and with WP control (*right*)
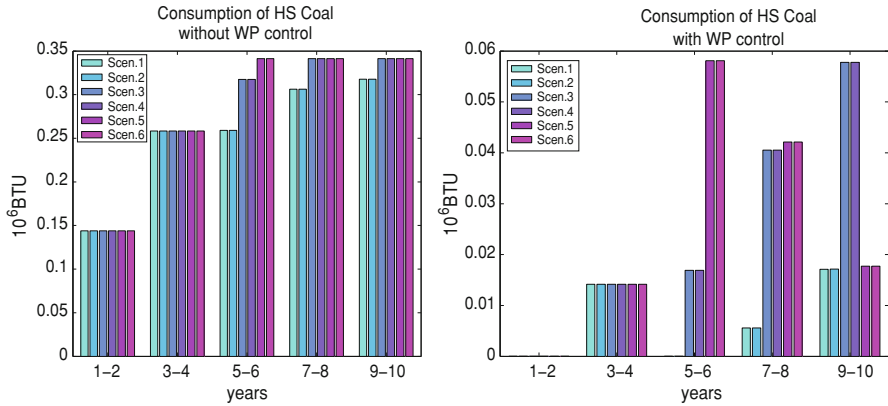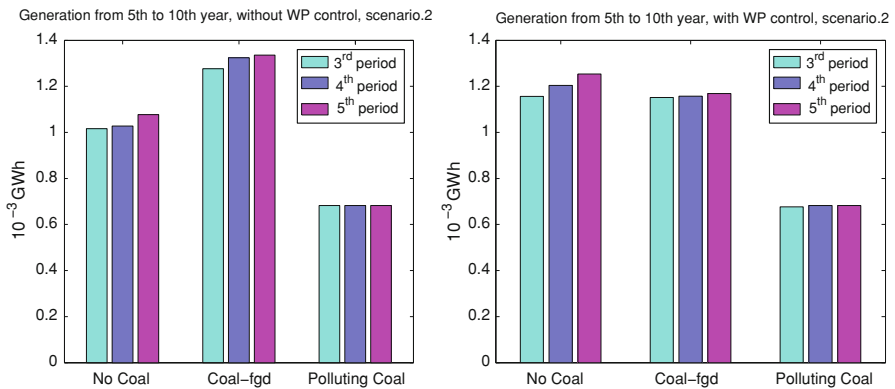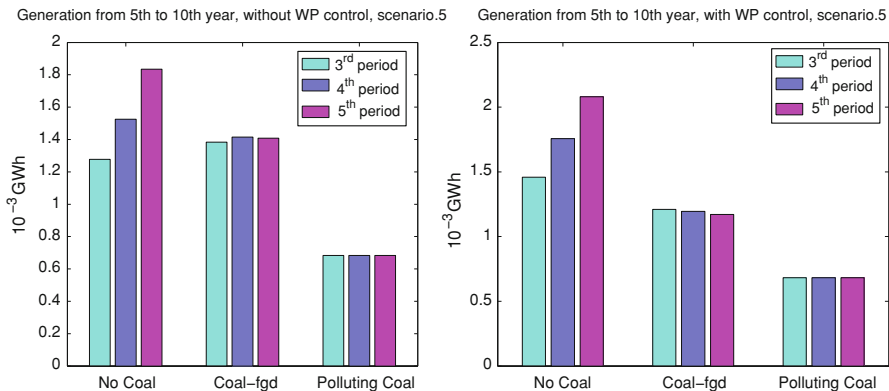
Fig. 3 HS coal consumption, without WP control (*left*) and with WP control (*right*)

Fig. 4 First stage generation, without WP control (*left*) and with WP control (*right*)

Fig. 5 Second stage generation, scenario 2, without WP control (*left*) and with WP control (*right*)

Fig. 6 Second stage generation, scenario 5, without WP control (*left*) and with WP control (*right*)

When comparing the two graphs of Fig. 3, we see that the right-hand side figure, corresponding to imposing WP control, shows a significant decrease in HS coal consumption: the corresponding values dropped from ranging between 0.14 and 0.34 to ranging between 0.0 and 0.06 millions of BTU (years 1–2 and 9–10, respectively, for both graphs).

As for the optimal expansion, during the first stage with WP control the model decides to build a peaker of 0.1 GW capacity, but without WP control it does not. During the second stage periods, there is no addition of capacity in either variant for scenario 2, the *easiest* one. For the *hardest* scenario 5, with WP control there is a significant addition of capacity in a gas and a peaker plants (0.3 and 0.1 GW, respectively). Without WP control, the peaker increases its capacity, but the gas plant capacity is increased by half of the amount decided by the variant with WP control. This explains the extra coal consumption of the WP control left-hand side graphs, when compared to the right-hand side ones in Figs. 2 and 3.

Figure 4 shows for each variant the optimal generation during the first stage years, discriminating the different types of plants: fgd-coal plants, polluting coal plants without fgd process, and other plants. We observe that without WP control, specially in the second period, there is an increase in the generation of fgd-coal plants. By contrast, with WP control, the WP constraint (16) reduces the fgd-coal generation in favor of other sources of energy (note the increase of other sources for the second period in the right-hand side of Fig. 4, when compared to the same column in the left-hand side).

Figures 5 and 6 show generation results for years 5–10 for scenario 2 and scenario 5, respectively. For scenario 2, generation from sources other than coal increases in about 20% with WP control, while polluting coal generation remains constant for both variants. As for fgd-coal generation, it drops in about 15% when there is WP control.

For scenario 5, there is an increase in about 20% on the generation of the no-coal plants with WP control (compare the first group of columns in the left- and right-hand graphs in Fig. 6). Generation of coal plants drops accordingly when there is WP control. As for the first stage, in both scenarios we see the effectiveness of the water pollution constraint (16) in reducing coal plants generations. Since scenario 2 is less stringent from the emissions point of view, the reduction is distributed by keeping

constant the generation of (the less expensive) polluting plants and substituting fgd-coal generation by other sources. By contrast, for the more stringent scenario 5, coal generation is clearly avoided, and the preference is given to other sources of energy.

Concluding remarks

We have described the structure and properties of a recently developed constrained bundle-type method suitable for solving some problems arising in energy applications, in particular those obtained through Lagrangian relaxation or Bender's decomposition. One example is the generation expansion planning problem with environmental constraints, for which numerical experiments have been reported. We note that already in our toy example, imposing environmental constraints, such as limits in the maximum allowed for sulfur emission of the coal mix and water pollution constraints, showed a perceptible impact on the expansion decisions and can be used as an effective tool to reduce pollution. For more sophisticated power systems, it can be interesting to analyze the impact of bundle compression and aggregation in the performance of the algorithm. For unconstrained problems, Hiriart-Urruty and Lemaréchal (1993, Ch. XIV, vol. II) contains a numerical comparison of the effect of compression and aggregation in bundle methods.

References

Avetisyan M, Bayless D, Gnuni T (2006) Optimal expansion of a developing power system under the conditions of market economy and environmental constraints. Energy Econ 28:455–466

Bienstock D, Shapiro J (1988) Optimizing resource acquisition decisions by stochastic programming. Manag Sci 34(7):215–229

Bonnans JF, Gilbert JCh, Lemaréchal C, Sagastizábal C (2006) Numerical optimization: theoretical and practical aspects, 2nd edn. Springer, Berlin

Campodónico N, Binato S, Kelman R, Pereira M, Tinoco M, Montoya F, Zhang M, Mayaki F (2003) Expansion planning of generation and interconnections under uncertainty. http://www.psr-inc.com.br/app/publicacoes.aspx#

Correa R, Lemaréchal C (1993) Convergence of some algorithms for convex minimization. Math Program 62:261–275

Fábián C (2000) Bundle-type methods for inexact data. Central Eur J Oper Res 8:35–55

Fábián C (2008) Handling CVaR objectives and constraints in two-stage stochastic models. Eur J Oper Res 191:888–911

Fletcher R, Leyffer S (1999) A bundle filter method for nonsmooth nonlinear optimization. Numerical Analysis Report NA/195, Department of Mathematcs, The University of Dundee, Scotland

Hiriart-Urruty J-B, Lemaréchal C (1993) Convex analysis and minimization algorithms. Number 305-306 in Grund. der math. Wiss. Springer (two volumes)

Karas E, Ribeiro A, Sagastizábal C, Solodov M (2009) A bundle-filter method for nonsmooth convex constrained optimization. Math Program 116:297–320

Kiwiel KC (1985) Methods of descent for nondifferentiable optimization. Lecture Notes in Mathematics, vol 1133. Springer, Berlin

Kiwiel KC (1991) Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. Math Program 52:285–302

Lemaréchal C, Nemirovskii A, Nesterov Yu (1995) New variants of bundle methods. Math Program 69:111–148

Marcato R, Sagastizábal C (2007) Introducing environmental constraints in generation expansion problems. Numer Linear Algebra Appl 14:357–368

Mifflin R (1982) A modification and extension of Lemaréchal's algorithm for nonsmooth minimization. Math Program Stud 17:77–90

Mifflin R (1996) A quasi-second-order proximal bundle algorithm. Math Program 73:51–72

Sagastizábal C, Solodov MV (2005) An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. SIAM J Optim 16:146–169

Schramm H, Zowe J (1992) A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. SIAM J Optim 2:121–152