On Lagrangian Decomposition for Energy Optimization

Claudia Sagastizábal*

Abstract. Real-life optimization problems often depend on data subject to unknown variations that can be due to imprecise measurements or to the stochastic nature of the data itself. When decisions need to be taken with high precision, it is important to employ methods that are reliable when subject to data variability. For complex problems such as those arising in the energy sector, advanced nonsmooth optimization techniques combined with Lagrangian decomposition provide a satisfactory answer to such concerns. We review recent approaches, including those referred to as having on-demand accuracy, for different Lagrangian functions. Throughout, the main concepts are illustrated by a simple example on optimal power management.

Mathematics Subject Classification (2000). Primary 90C56; Secondary 65K05, 49M27.

Keywords. Optimization, Nonsmoothness, Lagrangian Functions, Decomposition Techniques, Bundle Methods.

1. A Motivating Example

Consider the problem faced by a company owing two power plants, one thermal and one hydraulic, when the manager needs to decide the generation plan for the next day. To supply the requested demand of energy, decisions must be taken every half hour in a manner that optimally combines the technological limitations of each plant.

To assist the manager, an optimizer builds a model, successively addressing several issues, listed below.

- Identification of a goal to be achieved, such as to "minimize the generation cost", or "maximize the revenue", or "minimize the risk of having a deficit of energy"; this involves optimization of a so-called objective function. No matter what the chosen goal is, generation costs always enter into play and, hence, have a strong impact in the decisions. Here arises a first difficulty for the manager. For thermal plants, generation costs are easy to determine, as they are related to burning some fuel (fossil, nuclear). But for hydroplants, assigning a cost to generation is less straightforward, because 'fuel"

*Research partially supported by Grant CNPq 303840/2011-0 and FAPERJ.

Proceedings of the ICIAM,
Beijing, China, 2015

is the stored water, which costs nothing. The manager employs a substitute pricing the future lack of water as the cost of the thermal energy that would be needed in the future if the reservoir happened to be depleted. Sound mechanisms to give a meaning to the price of water require solving complex stochastic optimization problems; see [19] and [22].

- Representation of the plants capacity and of physical laws describing how each technology (thermal, nuclear, hydraulic) generates electricity; these are examples requiring definitions of constraint functions.

- Satisfaction of demand constraint: every time we enter a dark room and switch on the light, we expect electricity to be "waiting" there and the room to be lit. This extremely crucial constraint is difficult to deal with, not only because at planning time the exact amount of (future) demand is unknown, but also because electricity cannot be stored (except for limited amounts), so any extra production is just lost.

To write down a simplified mathematical formulation for this problem, consider that the electricity demand $d \in \Re$ is known, given as data. For thermal (hydraulic) generation variables p_T (p_H) with respective generation costs and technological constraint sets denoted by C_T and $\mathcal{P}_T$ (C_H and $\mathcal{P}_H$), the manager solves the optimization problem below:

$$v(d) := \left\{ \begin{array}{ll} \min & C_T(p_T) + C_H(p_H) \\ \text{s.t.} & p_T \in \mathcal{P}_T \, , p_H \in \mathcal{P}_H \\ & p_T + p_H = d \, , \end{array} \right. \tag{1}$$

where we denoted the optimal value $v(d)$ to emphasize the dependence of the output on the given demand. Since this data is an estimate, determining the impact of demand variations is often a concern for the company. To quantify such an impact, the manager can use *Lagrangian relaxation*. Specifically, associating a multiplier $x \in \Re$ to the demand constraint gives the Lagrangian

$$L(p_T, p_H, x) := C_T(p_T) + C_H(p_H) + x(d - p_T + p_H) \, .$$

If (1) is a convex problem, by duality arguments,

$$\begin{array}{ll} \min \\ p_T \in \mathcal{P}_T \\ p_H \in \mathcal{P}_H \end{array} \quad \begin{array}{l} \max \\ x \in \Re \end{array} \; L(p_T, p_H, x) = \quad \begin{array}{l} \max \\ x \in \Re \end{array} \quad \begin{array}{l} \min \\ p_T \in \mathcal{P}_T \\ p_H \in \mathcal{P}_H \end{array} \; L(p_T, p_H, x)$$

(without convexity, the equality is replaced by "$\geq$".) The optimal value of the rightmost problem, called *dual* to (1), equals $v(d)$ and has the form

$$v(d) = \max_{x \in \Re} \left\{ xd + \left(\begin{array}{ll} \min & C_T(p_T) - xp_T \\ \text{s.t.} & p_T \in \mathcal{P}_T \end{array} \right) + \left(\begin{array}{ll} \min & C_H(p_H) - xp_H \\ \text{s.t.} & p_T \in \mathcal{P}_H \end{array} \right) \right\} . \tag{2}$$

When seen as a function of d, and for each fixed dual variable x, the argument in the maximum is the sum of a linear term (with slope x), and a constant term (the

two minima over p_T and p_H). As a result, the function $v(d)$ is convex piecewise affine, and, by Danskin's theorem [10, Ch.VI.4.4], any $\bar{x}$ solving (2) satisfies

$$v(d') \geq v(d) + \bar{x}(d' - d) \quad \text{for all } d' \in \Re\,.$$

Since $v(d)$ is the cost for the company to satisfy the given demand, this inequality tells the manager that if demand was underestimated in, say, 1%, for the plants to produce more than the scheduled amounts $\bar{p}_T$ and $\bar{p}_H$, the company would have to spend at least additional $.01\bar{x}d$. Also, if the manager was to set the unit selling price below $\bar{x}$, the company would lose money.

Solving the dual problem (2) gives the manager the marginal cost $\bar{x}$, which can be used as a guide when setting the company selling prices. The interpretation of Lagrange multipliers as marginal costs (or shadow prices in the parlance of Linear Programming) has plenty of useful applications. Suppose now the company has three power plants (2 thermal, one hydraulic) and needs to keep carbon emissions below a threshold `max`, imposed by the government. This incorporates in (1) an "environmental" constraint of the form

$$\texttt{Pow2CO}_2(p_{T,1} + p_{T,2}) \leq \texttt{max}\,,$$

where $p_{T,i}$ is the thermal generation of plant $i = 1, 2$ and $\texttt{Pow2CO}_2$ is a scalar converting thermal power into carbon emissions. In this case, the corresponding optimal Lagrange multiplier gives an indication of the value of *green certificates*, delivered by the government to the company as a compensation for respecting the pollution limitations.

Our simple example (1) illustrates the importance of computing with high accuracy optimal dual variables: changing electricity prices in the cents has a tremendous socio-economical impact. For simple problems, $\bar{x}$ can be obtained directly from solving (1). For more realistic problems, the direct method is not possible and the dual approach (2), separating the optimization problem over $\mathcal{P}_T$ from the one over $\mathcal{P}_H$, is the only one applicable. The reason is that for realistic problems the sets $\mathcal{P}_T$ and $\mathcal{P}_H$ describe very different technologies, and no optimization package can solve the problem directly (typically, $\mathcal{P}_T$ uses 0-1 variables and $\mathcal{P}_H$ is described by nonconvex constraints). In (2), by contrast, the optimization problem in variable p_T is *decoupled* from the one in variable p_H, and separate off-the-shelf solvers can be employed for their respective solution. Of course, the price to pay is that now there is one more optimization layer, the maximization over the dual variable x.

The remainder of this work is organized as follows. In Section 2 we describe typical problems arising in Energy Optimization. Section 3 presents the main ingredients of nonsmooth optimization methods that can be used to maximize the dual function in (2), including bundle methods [2]. Section 4 discusses when the dual solution provides primal minimizers for Lagrangians different from the classical one, used to derive (2). In Section 5, various approximations for those Lagrangians are examined under the light of a recent variant of bundle methods, able to deal with inexact oracle information.

Our notation is fairly standard. Throughout we use the Euclidean inner product $\langle x, p\rangle = x^\top p$ for two column vectors x and p, and denote the induced Euclidean norm by $\|\cdot\|$. For a set S, its cardinality and convex hull are respectively denoted by $|S|$ and $\operatorname{conv} S$.

2. Some Energy Models

In a real-life setting, the optimal power management problem (1) involves several power plants (thermal with fossil fuel, nuclear, aeolian, a set of hydro-plants along a river, etc), with very different technological sets. Moreover, the joint management of the power mix is done along certain time horizon, discretized in many time steps. Because of interconnections between different electrical networks (of cities in a country, or more generally, of countries in a region), electricity can be brought from far away if there is need, or, reciprocally, excessive generation can be sent somewhere else through the network. This flexibility makes it possible to combine efficiently units having strict generation rules but low costs with more expensive units that are able to produce electricity "as soon as required". In a manner similar, generation costs can be reduced by exchanging energy between regions with different weather conditions or with shifted load peaks.

To fix the notation and setting we recall below the main features of Example 3.1 from [22]. Suppose the time horizon is composed of periods $\{1,\ldots,T\}$ and there is a total of I units in the mix. If p_i^t denotes the energy produced by the i-th unit during the period t, and $p_i = (p_i^1, p_i^2, ..., p_i^T)$ s the vector with the generation of unit i for the whole horizon of time, the decision vector is $p = (p_1, \ldots, p_i, \ldots, p_I)$. The generation cost is usually separable by units

$$C(p) = \sum_{i=1}^{I} C_i(p_i)$$

for some individual convex cost functions C_i, which may be further separable by time steps: $C_i(p_i) = \sum_{t=1}^{T} C_i^t(p_i^t)$.

As for the constraints, they typically are of *dynamic* or of *static* nature. The set of dynamic constraints D includes all the operating rules for each unit, and often couples variables of the same unit along time steps:

$$D = \prod_{i=1}^{I} \mathcal{P}_i \quad \text{with } p_i \in \mathcal{P}_i \text{ for } i = 1\,,\ldots,\,,I\,.$$

Describing the dynamic feasible sets $\mathcal{P}_i$ is often a complex matter, that may involve nonlinear relations and 0-1 variables, to represent the technological constraints of thermal, aeolian, nuclear, hydro units.

Static constraints refer to relations like satisfaction of demand or pollution limits, that couple the generation of all the plants at each time step:

$$S = \prod_{t=1}^{T} S^t \quad \text{with } (p_1^t, p_2^t, \ldots, p_I^t) \in S^t \text{ for } t = 1\,,\ldots,\,,T\,.$$

These are usually affine relations, such as

$$\sum_{i=1}^{I} p_i^t = d^t\,, \quad \text{for } t = 1, \ldots, T\,,$$

where d^t is the demand at time t. For some models, demand is uncertain and depends on some random variable, and in this case both the static feasible sets and generation variables vary with uncertainty.

Similar affine static constraints can be written to include shortages and energy interchanges between subsystems of the mix. Transmission constraints (bounding the capacity of each arc in the network), and security constraints (demand is satisfied even after the outage of one line) are also static constraints. With a DC power flow model, such constraints are affine, but also numerous, because the network has several hundreds of nodes. When setting security constraints, there will be several hundred thousand linear constraints at each time step.

Reserve constraints, which are natural inequalities, are often written as equalities by introducing slack variables; these variables can be interpreted as the generation of a fictitious unit, possibly with nonzero operating cost. To ensure feasibility, demand constraints can also incorporate slack variables representing unsupplied energy, usually penalized with a high deficit cost.

A generic formulation for the optimal power management problem is:

$$\left\{ \begin{array}{ll} \min\limits_{p} & C(p) \\ \text{s.t.} & p \in S \cap D\,. \end{array} \right. \tag{3}$$

Lagrangian relaxation is a convenient tool for this type of problems. Suppose only demand satisfaction defines the static set. Then, the multiplier

$$x = (x^1, \ldots, x^t, \ldots, x^T) \in \Re^T$$

gives a Lagrangian

$$L(p, x) = \sum_{i=1}^{I} C_i(p_i) + \left\langle x, d - \sum_{i=1}^{I} p_i \right\rangle$$

that is separable along units:

$$L(p, x) = \sum_{i=1}^{I} L_i(p_i, x) + \langle x, d \rangle \qquad \text{for } L_i(p_i, x) = C_i(p_i) - \langle x, p_i \rangle\ . \tag{4}$$

The corresponding dual function $\theta(x) = \max_{p \in \mathcal{P}} L(p, x)$ inherits separability

$$\theta(x) = \sum_{i=1}^{I} \theta_i(x) + \langle x, d \rangle$$

where

$$\theta_i(x) = \min_{p_i \in \mathcal{P}_i} L_i(p_i, x) = \left\{ \begin{array}{ll} \min & C_i(p_i) - \langle x, p_i \rangle \\ \text{s.t.} & p_i \in \mathcal{P}_i\,. \end{array} \right. \tag{5}$$

A nice interpretation of the dual approach follows from the role played by x is a price. The negative of each $\theta_i(x)$ is an optimization problem in which the ith plant maximizes its benefit, given that the generated power is remunerated at unit price x (as in (8) below).

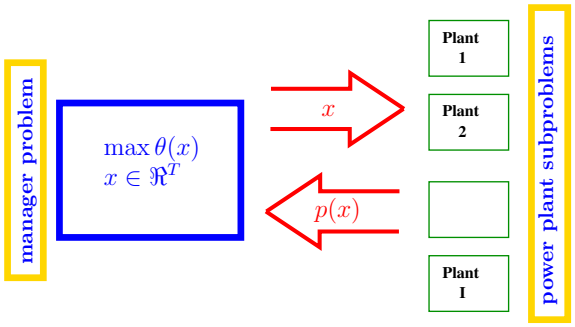

Figure 1. Price Decomposition

Figure 1 represents such *price decomposition* scheme, in which the manager solves a generalization of problem (2) (that is $\max_{x\in\Re^T} \theta(x)$), by applying an iterative process. Specifically,

- as illustrated by the top arrow in Figure 1, the manager proposes to pay the power plants a unit price x for their production.
- Knowing the remuneration, the ith power plant determines the production maximizing its benefit, i.e.,

$$\text{finds } p_i(x) \in \mathcal{P}_i \text{ such that } \theta_i(x) = L_i(p_i(x), x). \tag{6}$$

  The solution $p_i(x)$ is one component of the bottom arrow in Figure 1, with the information submitted to the manager: *if you pay me x, I'm willing to generate $p_i(x)$ power.*
- Once the output of all the power plants is available, the manager checks if the total generation is enough to satisfy the demand. If not, a new price x is proposed to the power plants, and the process is repeated.

In this mechanism, the manager needs to maximize the nonsmooth dual function θ. We now explain the basic ingredients needed to put in place efficient algorithms to solve the manager problem with high accuracy.

3. A Primer on Nonsmooth Optimization

Since optimization algorithms are usually designed for solving minimization problems, from now on we shall work with the negative of the functions from (5).

Instead of solving (3) directly, our manager applies a decomposition method to solve the convex dual problem

$$\min_{x\in\Re^T} f(x) \quad \text{for } f(x) = -\theta(x)\,. \tag{7}$$

The left block in Figure 1 representes the calculations needed for one iteration to solve (7). We now review some notions and methods for solving this problem, for full details we refer the reader to [2, Part II].

3.1. The oracle. At each iteration of the decomposition process, to decide which price to send to the plants the manager uses available information, provided by an *oracle* or *black-box*. In Figure 1, this wording refers to the output of each individual plant $i = 1, \ldots, I$ (the right squares in the figure). The mathematical formulation of the i-th oracle amounts to evaluating the convex subfunction

$$f_i(x) := \max_{p_i\in\mathcal{P}_i} -L_i(p_i, x) = \left\{ \begin{array}{ll} \max & \langle x, p_i\rangle - C_i(p_i) \\ \text{s.t.} & p_i \in \mathcal{P}_i \end{array} \right. \tag{8}$$

at the given price x, sent by the manager. This revenue maximization problem is solved by finding one $p_i(x)$ attaining the maximum. When more than one maximizer $p_i(x)$ exists, the subfunction is nonsmooth.

The graphical representation of a typical function f_i in Figure 2 shows with dotted lines several of the affine functions $-L_i(p_i, \cdot)$, for different $p_i \in \mathcal{P}_i$. In particular, the lines for p_i^1 and p_i^2 illustrate a common phenomenon of nonsmooth optimization: f_i fails to be differentiable on a set with null measure, but the minimum is precisely in this set. In the figure, $f_i(\bar{x}) = -L_i(p_i^1, \bar{x}) = -L_i(p_i^2, \bar{x})$, and, hence, either p_i^1 or p_i^2 could be taken as $p_i(\bar{x})$ in (6), or in its equivalent formulation using (8).

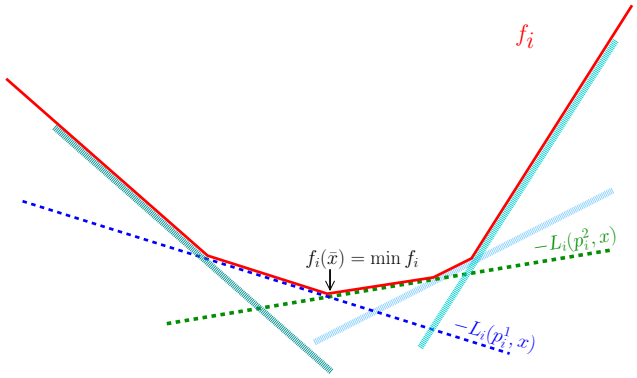

Figure 2. Piecewise affine functions are often minimized at a kink

The Convex Analysis subdifferential of the function f_i at x is given by the maximizers in (8), see [10, Ch.VI]:

$$\partial f_i(x) = \text{conv}\Big\{p_i(x) : \text{ for } p_i(x) \in \mathcal{P}_i \text{ such that } f_i(x) = -L_i(p_i(x), x)\Big\}\,.$$

Since the manager problem has the form

$$\min_{x\in\Re^T} f(x) \quad \text{for } f(x)=\sum_{i=1}^{I} f_i(x) - \langle x, d\rangle \ ,$$

it follows that evaluating each subfunction f_i at a given x provides for free one subgradient. Namely, having $p_i(x)$ for $i=1,\ldots,I$ yields both $f(x)$ and

$$g(x)=\sum_{i=1}^{I} p_i(x) - d \in \partial f(x)\,. \tag{9}$$

We now describe different methods of employing the oracle information to define a sequence $\{x^k\}$, the successive prices sent by the manager to the power plants.

3.2. Subgradient Algorithms. By convexity, the minimizers of f in problem (7) are fully characterized by the Fermat optimality condition

$$0\in\partial f(\bar{x}) = \mathrm{conv}\Big\{p(\bar{x})\Big\} - d\,.$$

To detect convergence, an algorithm checks approximate satisfaction of this condition. For instance, the method stops if, for an approximate subgradient $\hat{g}^k\in\Re^T$ and an error $\varepsilon_k\geq 0$ built along iterations, it holds that

$$\hat{g}^k\in\partial_{\varepsilon_k} f(\hat{x}^k) \quad \text{with} \quad \|\hat{g}^k\| \text{ and } \varepsilon_k \text{ sufficiently small,} \tag{10}$$

where $\partial_\varepsilon f$ is the ε-subdifferential in Convex Analysis, see [10, Ch.XI].

Endowing an algorithm with a stopping criterion is of fundamental importance. However, the first nonsmooth optimization method considered here, the *subgradient algorithm* [23], overlooks this crucial issue. The method, called Uzawa's in optimal control, has a very simple updating rule.

More precisely, having the oracle information $f(x^k)$ and $g(x^k)$ provided by the output $p_i(x^k)$ of the power plants for $i=1,\ldots,I$, the manager defines the next price as

$$x^{k+1}=x^k - t_k g(x^k) \text{ for a suitable stepsize } t_k>0,$$

satisfying the conditions $\sum t_k=\infty$ and $t_k\to 0$; see [2, Ch.9.3.1].

This type of methods, which can be declined in several variants for the stepsize (all satisfying the conditions above), is very popular because the update is easy to implement. Typically, subgradient methods stop after reaching a maximum number of iterations (when the manager gets tired?). As such, they are not suitable when high precision is required as it is the case if the manager needs, for example, to set a price for the generated electricity.

Another drawback of subgradient methods is that the sequence of function values is not monotone: $f(x^{k+1})$ can be larger than $f(x^k)$. The reason is that, contrary to the smooth case and no matter how small the stepsize t_k is taken, the direction opposed to a subgradient may not provide descent.

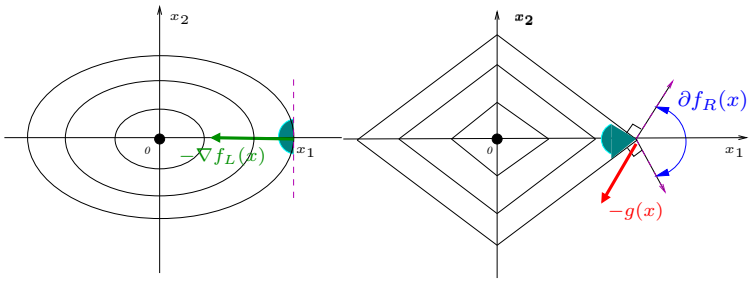

Figure 3. A subgradient may not provide descent

Figure 3, taken from [2], illustrates this phenomenon, displaying the level-lines for two functions minimized at $0 \in \Re^2$; the differentiable function $f_L(x_1, x_2) = x_1^2 + 2x_2^2$ on the left and the nondifferentiable function $f_R(x_1, x_2) := |x_1| + 2|x_2|$ on the right. The shadowed area shows all descent-directions in the two cases. Observe that for this simple case, the direction opposite to $g(x) = (1, 2) \in \partial f_R(0, x_2)$, for $x_2 > 0$, is **not** a direction of descent.

Figure 4 shows a typical trajectory for a sequence of function values generated with the subgradient method. We observe that $f(x^4)$, $f(x^5)$, $f(x^6)$ all have larger values than $f(x^3)$ and that the condition $t_k \to 0$ "shortens" the improvement toward a minimizer as iterations progress. This zigzagging phenomenon slows down the convergence speed and makes it difficult, if not impossible, to achieve high accuracy with this type of methods.

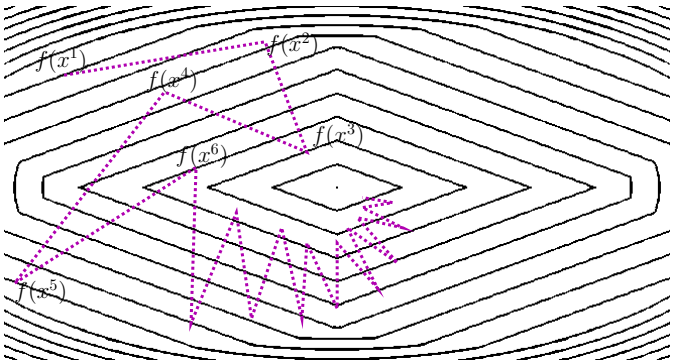

Figure 4. Zigzagging of a subgradient method

3.3. Cutting-plane Methods. The subgradient method update does not make use of the function information $f(x^k)$, provided by the oracle together with the gradient $g(x^k)$. By contrast, cutting-plane methods, [3] and [12], use all the oracle information to define *lower linearizations*:

$$f(x) \geq f(x^k) + \langle g(x^k), x - x^k \rangle \quad \text{for all } x \in \Re^T,$$

and build a convex *model* M^k for the function f. For the manager problem, (4) gives linearizations of the form

$$f(x^k) + \langle g(x^k), x - x^k \rangle = -\sum_{i=1}^{I} L_i(p_i(x^k), x) - \langle x, d \rangle \ . \tag{11}$$

The interest of having a model lies in the fact that it can be used as a replacement for the unknown function f to compute the next iterate:

$$x^{k+1} \text{ solves } \min_{x \in \Re^T} M^k(x) \text{ where } M^k(x) := \max_{j \le k} \left\{ f(x^j) + \langle g(x^j), x - x^j \rangle \right\} .$$

This updating rule is more involved than the subgradient method update, as it requires solving a linear programming problem (by adding an extra variable and constraints, to represent the maximum defining the model M^k, as in Section 5.1). The additional computational effort is compensated by the availability of a stopping test. Indeed, the distance between the model and the function gives an optimality measure

$$\text{in cutting-plane methods, the nominal decrease } \delta^{k+1} := f(x^{k+1}) - M^k(x^{k+1}) \to 0$$

as $k \to \infty$; see [2, Ch.9.3.2]. Notwithstanding, for this convergence result to hold, **all** the past linearizations must be kept in the model. As k grows, the linear program defining iterates has more and more constraints, many of them similar. The optimization problem becomes ill conditioned, and the algorithm struggles to achieve high accuracy. A way out of this tailing-off effect is to *clean the model*, and eliminate *inactive* constraints, for example defining

$$M^k_{act}(x) := \max_{j \in J^k_{act}} \left\{ f(x^j) + \langle g(x^j), x - x^j \rangle \right\}$$

for the active index set

$$J^k_{act} := \left\{ j \le k : M^k(x^{k+1}) = f(x^j) + \langle g(x^j), x^{k+1} - x^j \rangle \right\} ,$$

and taking

$$\begin{aligned} M^{k+1}(x) &= \max \left\{ M^k_{act}(x), f(x^{k+1}) + \langle g(x^{k+1}), x - x^{k+1} \rangle \right\} \\ &= \max \left\{ f(x^j) + \langle g(x^j), x - x^j \rangle : j \in J^{k+1} = J^k_{act} \cup \{k+1\} \right\} . \end{aligned}$$

Unfortunately, by the nature itself of the approach, dropping inactive cuts destroys the convergence theory of cutting-plane methods. We shall see that the bundle methods considered below address this issue maintaining the convergence results, thanks to a technique referred to as *bundle compression*.

The consideration of a model endows the cutting-plane approach with a stopping test. But the lack of monotonicity in the successive function values remains. The method exhibits an inherent *instability* that leads to bad numerical behavior near an optimum, preventing once more reaching high accuracy. When getting close to a solution, $g(x^k) \approx 0$, and "horizontal" affine functions are added to the model. This makes iterates move far away from the set of minima and oscillate wildly; we refer to [2, Ch.9.3.2] for detailed explanations; see also [10, Vol.II].

3.4. Bundle Methods. The key ingredient to defeat instability and oscillations is monotonicity. To the important concept of model, that gives a stopping test, bundle methods add the notion of *serious steps*. The serious step subsequence $\{\hat{x}^k\}$ gathers iterates that satisfy a *descent* rule, depending on a parameter $m \in (0,1)$:

$$\text{Set } \hat{x}^{k+1} = x^{k+1} \text{ if } f(x^{k+1}) \leq f(\hat{x}^k) - m\delta^k, \text{ for } \delta^k = f(\hat{x}^k) - M_k(x^{k+1})\,. \quad (12)$$

Otherwise, the serious step is maintained: $\hat{x}^{k+1} = \hat{x}^k$ and the step is declared *null*. It can be shown that $0 \leq \delta^k \to 0$, and that the limit points of the monotone serious-step subsequence $\{\hat{x}^k\}$ minimize f; see [2, Ch.10.3.4]. The convergence analysis also shows that if there are only finitely many serious steps, then the last one is a solution and the infinite tail of null steps that follows also converges to the last serious step.

To address the oscillating phenomenon in the cutting-plane method, the current serious-step is used as a stabilization center in the rule defining the next iterate. One possibility, called *proximal bundle method*, takes

$$x^{k+1} \text{ solves } \min_{x\in\Re^T} M^k(x) + \frac{1}{2t_k}\|x - \hat{x}^k\|^2\,, \quad (13)$$

for a proximal stepsize $t_k > 0$. Other bundle variants, respectively called with trust-region and level bundle, minimize the cutting-plane model over a ball around $\hat{x}^k$, or find the closest point to $\hat{x}^k$ that reduces sufficiently the cutting-plane model; see [2, Ch.10.2].

In bundle methods the choice of the model M^k is rather flexible. Incidentally, the word *bundle* refers to the collection of past information that is used to define the model, which can be nonpolyhedral. Or, as explained, it can be a cutting-plane model, which makes (13) a quadratic programming problem.

In all cases, to prevent the size of (13) from growing indefinitely as with the cutting-plane method, the bundle of information can be reduced to the active linearizations only, or it can be *compressed*, using certain aggregate linearization. To define this linearization, notice that that convexity of M^k makes (13) strongly convex, with unique solution characterized by the optimality condition

$$0 \in \partial M^k(x^{k+1}) + \frac{1}{t_k}(x^{k+1} - \hat{x}^k)\,,$$

and, hence,

$$x^{k+1} = \hat{x}^k - t_k\hat{g}^k \quad \text{for } \hat{g}^k \in \partial M^k(x^{k+1})\,. \quad (14)$$

After solving (13), the *aggregate linearization*, defined by

$$\ell_k(x) = M^k(x^{k+1}) + \left\langle \hat{g}^k, x - x^{k+1}\right\rangle\,,$$

can be computed. For the next iteration, **any convex model** satisfying

$$\max\left\{\ell_k(x), f(x^{k+1}) + \left\langle g(x^{k+1}), x - x^{k+1}\right\rangle\right\} \leq M^{k+1}(x) \leq f(x)\,, \quad (15)$$

can be taken (keeping in mind that if M^k is not polyhedral, (13) will no longer be a quadratic program). With the leftmost choice, in particular, (13) is a quadratic program with only two constraints: $|J^{k+1}| = 2$. The corresponding bundle method is sometimes called *poor man's*. While still convergent, this variant is slower than others using richer models (with larger sets J^{k+1}). For combinatorial problems, the poorman bundle algorithm has been successfully applied as an improvement of the volume method; see [1]. For some other applications, it may preferable to use a better model, spend more time in each iteration and make more progress, thus reaching a good accuracy in less time overall.

Like with cutting-plane methods, convergence depends on showing that $\delta^k \to 0$. Working out the algebra it can be seen that

$$\hat{g}^k \in \partial_{\varepsilon_k} f(\hat{x}^k) \quad \text{for } \varepsilon_k \geq 0 \text{ and } \delta^k = t_k \|\hat{g}^k\|^2 + \varepsilon_k \,.$$

In addition to the possibility of reducing or compressing the bundle (which keeps small the size of (13)), the relation above is a second strong point in favor of bundle methods: by driving δ^k to zero, the method ensures eventual satisfaction of the approximate stopping test (10).

4. What Can The Manager Do With The Result?

The power plant manager applied the decomposition scheme illustrated by Figure 1 because solving the problem directly, as a whole, was not possible. Such a detour through the dual function and its maximization with some nonsmooth algorithm has an impact that we explain below.

4.1. The Struggle for Feasibility. When the manager uses a bundle method (the best available option) to define prices, the output of the iterative process is

$$\hat{g}^{klast} \approx 0 \quad \text{and} \quad \varepsilon_{klast} \approx 0 \,,$$

where *klast* denotes the iteration triggering the stopping test.

Suppose a polyhedral cutting-plane model is used:

$$M^k(x) := \max_{j \in J^k} \left\{ f(x^j) + \left\langle g(x^j), x - x^j \right\rangle \right\}$$

for some index set J^k of past iterations. Then $\partial M^k(x^{k+1}) = conv\{g(x^j) : j \in J^k_{act}\}$ and, hence, there exists a simplicial vector $0 \leq \alpha^k \in \Re^{|J^k|}$ with $\sum_{j \in J^k} \alpha^k_j = 1$ such that the aggregate gradient in (14) satisfies

$$\hat{g}^{klast} = \sum_{j \in J^{klast}} \alpha^{klast}_j g(x^j) \,.$$

In view of (9), this amounts to having

$$\sum_{i=1}^{I} p^{conv}_i \approx d \qquad \text{for } p^{conv}_i := \sum_{j \in J^{klast}} \alpha^{klast}_j p_i(x^j) \,,$$

and where each $p_i(x^j) \in \mathcal{P}_i$, by construction. If the manager is interested in pricing the electricity generation, i.e., in obtaining an accurate x^{klast}, the output provided by the decomposition approach is highly satisfactory, as the bundle method ensures the method ended with a reliable stopping criterion, finding a solution with high accuracy.

However, if the manager is also interested in determining a production plan, some additional calculations might be needed. For some power plants, typically thermal ones, the set $\mathcal{P}_i$ is not convex and, hence,

$$\text{for all } j \in J^{klast} \text{ each } p_i(x^j) \in \mathcal{P}_i \text{ but } p_i^{conv} \notin \mathcal{P}_i\,.$$

This means that the outcome of the decomposition approach cannot be sent directly as a directive to the power plants: if $\mathcal{P}_i$ involved binary variables to represent units switched on and off, the convex combination p_i^{conv} might require to turn on only a fraction of the corresponding unit!

A primal point associated with $\hat{x}^{klast}$ is $p(\hat{x}^{klast})$, the information sent by the oracle solving the power plant subproblems. Unlike p_i^{conv}, each $p_i(\hat{x}^{klast}) \in \mathcal{P}_i$, even when $\mathcal{P}_i$ is not convex. However, and contrary to p_i^{conv},

$$\sum_{i=1}^{I} p_i(\hat{x}^{klast}) \not\approx d\,.$$

After all the efforts putting in place a decomposition scheme, the manager ends the process with a feasible generation plan which does not satisfy demand, or with a plan of electricity generation that satisfies the demand, but cannot be produced by the power plants of the company. Recovering primal feasibility without losing much optimality is the subject of many works in energy optimization, we refer to [4], [7], [29], and the recent review on unit-commitment, [25].

For the dual approach to have any chance to provide primal points that are satisfactory, at the very least the distance between the primal and dual optimal values, such as (3) and (7), or (1) and (2), should be the same. This distance is called *duality gap* [10]; see also [14].

For the Lagrangian in (4), the duality gap can be shown to be zero only for problems that are convex if certain *constraint qualification* [11, Ch.1] condition holds. In this case, the error ε_k computed by the bundle method measures the optimality gap; see [16, Rem.4.4]. For nonconvex problems, the duality gap is positive, so if the manager needs not only the electricity price but also a generation plan, there are two possibilities:

- After solving (7), initiate a *primal recovery* phase, ideally using p^{conv} as a guide and without losing the separability structure; see [5], [7].

- Primal recovery is based on heuristic approaches that (try to) generate primal solutions, even for discrete or non convex feasible sets. In such a setting, not much can be shown theoretically. At least not with the classical Lagrangian (4): we shall see that a different Lagrangian can close the duality gap, even for nonconvex problems (the price to pay will be a loss of separability).

Table 1 gives in a condensed form the main advantages and drawbacks of using the classical Lagrangian relaxation approach presented so far.

Table 1. Classical Lagrangian for Nonconvex Problems (3)

Lagrangian definition	Subproblem Separability	Shadow Price	Primal Feasibility	Zero Duality Gap
(4)	YES	YES	NO	NO

4.2. Exploring Alternative Lagrangians. Since problem (3) is too hard to tackle directly, separability was induced by solving the dual problem derived from (4). As dual approaches solve in fact the bi-dual of the initial primal problem, when (3) is nonconvex the decomposition technique ends up with a positive duality gap, which can be thought of as the price to be paid for separability.

The classical Lagrangian (4) is a particular case of certain general dualizing parameterizations, described in Chapter 11.K in [21], devoted to augmented Lagrangians and nonconvex duality. Given two dual variables, $x \in \Re^T$, as before, and $0 \leq r \in \Re$, we shall consider two of such options:

the *Proximal* Lagrangian from [21, Ex.11.57]
$$L^{prox}(p,x,r) = L(p,x) + \frac{r}{2}\sum_{t=1}^{T}\Big(\sum_{i=1}^{I} p_i^t - d^t\Big)^2 \quad (16)$$

and

the *Sharp* Lagrangian from [21, Ex.11.58]
$$L^{\#}(p,x,r) = L(p,x) + r\sum_{t=1}^{T}\Big|\sum_{i=1}^{I} p_i^t - d^t\Big|. \quad (17)$$

Sharp Lagrangians can be defined for any norm, the expression above uses the ℓ_1-norm, taking just the absolute value of each t-component of the relaxed constraint, for convenience. By (4), the corresponding (negative of the) dual functions are

$$f^{prox}(x,r) = \max_{p\in D} -\sum_{i=1}^{I} L_i(p_i,x) - \frac{r}{2}\sum_{t=1}^{T}\Big(\sum_{i=1}^{I} p_i^t - d^t\Big)^2$$

and

$$f^{\#}(x,r) = \max_{p\in D} -\sum_{i=0}^{I} L_i(p_i,x) - r\sum_{t=1}^{T}\Big|\sum_{i=1}^{I} p_i^t - d^t\Big|,$$

where we used the relation $\min(-L) = -\max L$.

When comparing the new dual functions with (8), we observe that in both cases the augmentation term kills separability along the i-components. For the scheme in Figure 1, this amounts to gathering all the separate squares on the right into one tall rectangle and, hence replacing the difficult (3) by a sequence of difficult problems! The tactics explained in Section 5 aim at providing a compromise in this sense.

The loss of separability is compensated by the elimination of the duality gap. Under mild assumptions [21, Thm.11.59] shows that

$$\text{the optimal value of} \quad \begin{array}{ll} \text{either} & \min\limits_{(x,r)\geq 0} f^{prox}(x,r) \\ \text{or} & \min\limits_{(x,r)\geq 0} f^{\#}(x,r) \end{array} \quad \text{equals the optimal value of (3),}$$

even when the primal problem is not convex.

Beware that closing the duality gap does not necessarily entail primal feasibility, although the feature is already a good starting point (without closing the gap there is no hope). The question of the extent to which a null duality gap gives a primal solution has its answer in the necessary and sufficient conditions for *exactness*, stated in [21, Thm.11.61]. This result involves the concepts of *proximal subgradient* for (16) and *calmness* for (17). Rather than detailing those technical concepts, we consider the different dual functions for an instance of (1), putting a particular emphasis on the Sharp Lagrangian.

Example. Suppose in (1) the data is

$$C_t(p_T) = p_T^2\,, C_H(p_H) = \frac{1}{2}p_H^2\,, \mathcal{P}_T = \{0, P\}\,, \mathcal{P}_H = [0, P]\,, d = \frac{3}{2}P\,.$$

For this simple problem, writing and solving the optimality conditions by hand gives that

$$\bar{p}_T = P\,, \quad \bar{p}_H = \frac{1}{2}P\,, \quad \bar{x} = P\,, \quad v(d) = \frac{9}{8}P^2\,.$$

The separable dual function (7) is the sum $f(x) = f_T(x) + f_H(x) - xd$. After some algebra, the two subfunctions defined in (8) have the expressions

$$f_T(x) = \max_{p_T \in \{0,P\}} \{xp_T - p_T^2\} = \max\Big(0, P(x-P)\Big)$$

and $f_H(x) = \max\limits_{p_H \in [0,P]} \{xp_H - \frac{1}{2}p_H^2\} = \frac{1}{2}\,proj_{\,[0,P]}(x)^2 + \max\Big(0, P(x-P)\Big)$, so the dual function is

$$f(x) = \frac{1}{2}\,proj_{\,[0,P]}(x)^2 + 2\max\Big(0, P(x-P)\Big) - xd\,.$$

Its minimizer $\bar{x} = P$ has a multiplier $\bar{\alpha} = \frac{3}{4} = d/2$, given by the the optimality condition. With this multiplier, the convex combinations of the primal points $p_T(\bar{x}) = P$ and $p_H(\bar{x}) = P$ result in $p^{conv} = \bar{\alpha}p_T(\bar{x}) + (1-\bar{\alpha})p_H(\bar{x})$, which satisfies the demand constraint (note than $\bar{\alpha}p_T(\bar{x}) = \frac{3}{4}P \notin \{0, P\}$). Finally, the optimal dual value is $f(\bar{x}) = -P^2$, so $\theta(\bar{x}) = -f(\bar{x})$ gives a duality gap equal to $\frac{1}{8}P^2$.

The computations for the proximal and sharp dual functions can be derived by rewriting the nonseparable problem, so that p_H is a function of p_T. To describe the procedure, consider the Lagrangian (17)

$$f^{\#}(x,r) = -xd + \left\{ \begin{array}{ll} \max & x(p_T + p_H) - p_T^2 - \frac{1}{2}p_H^2 - r\Big|p_T + p_H - d\Big| \\ \text{s.t.} & p_T \in \{0, P\} \\ & p_H \in [0, P]\,, \end{array} \right.$$

and its equivalent problem $f^{\#}(x,r) = -xd + \max\left\{F^{\#}(x,r,p) : p = 0, P\right\}$ for the functions

$$F^{\#}(x,r,p) = xp - p^2 + \left\{ \begin{array}{ll} \max & xp_H - \frac{1}{2}p_H^2 - r\Big|p + p_H - d\Big| \\ \text{s.t.} & p_H \in [0,P]\,. \end{array} \right.$$

By the optimality conditions, each of these functions has a unique solution parameterized by p, denoted by $q^{\#}(p)$ and given by

$$q^{\#}(p) = proj\ _{[0,P]}\Big(x - r\, proj\ _{[-1,1]}\Big(\frac{x+p-d}{r}\Big)\Big)\,.$$

Replacing this value in $F^{\#}$ and solving over the two choices $p = 0$ and $p = P$ yields the sharp function whose level sets are displayed on the left graph in Figure 5.

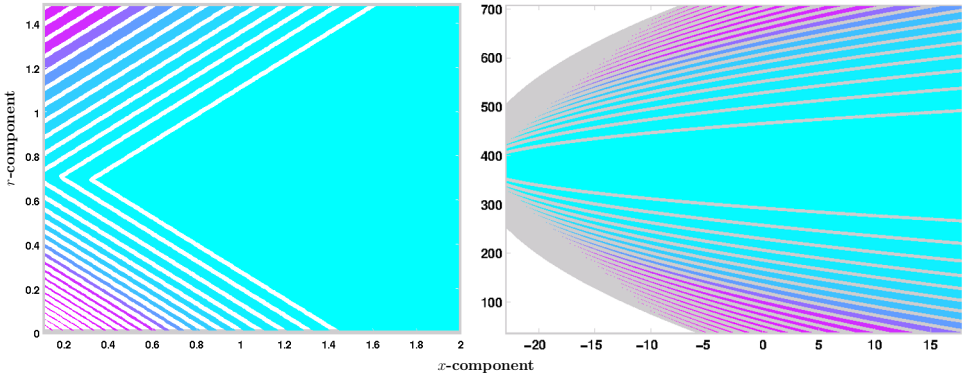

Figure 5. Level sets for a sharp and proximal dual function (left and right)

The minimization of the sharp dual function $f^{\#}$ gives an optimal value equal to $\frac{9}{8}$ (no duality gap), with minimizers $\bar{x}^{\#} = P$ and $r \geq \bar{r}^{\#} = \frac{1}{2}P$. The primal points corresponding to $\bar{x}^{\#}$ and $\bar{r}^{\#}$ are $\bar{p}_T^{\#} = P$ and $\bar{p}_H^{\#} = q^{\#}(P) = \bar{x}^{\#} - \bar{r}^{\#} = \frac{1}{2}P$, which satisfy all the primal constraints, including demand. Table 2 adds to Table 1 the new information.

Table 2. Classical and Sharp Lagrangian for Nonconvex Problems (3)

Lagrangian definition	Subproblem Separability	Shadow Price	Primal Feasibility	Zero Duality Gap
(4)	YES	YES	NO	NO
(17)	NO	YES	YES	YES

Proceeding likewise for the proximal Lagrangian (16), we obtain that

$$q^{prox}(p) = proj\ _{[0,P]}\Big(\frac{x}{1+r} - r\frac{p-d}{1+r}\Big)\,;$$

the specific calculations for this Lagrangian are left to the reader. The right graph in Figure 5 shows the level sets of the corresponding (smooth) proximal dual function.

5. Designing Alternative Bundle Methods

For the manager problem (1), or its instance in Section 4.2, combining (4) with (11) gives the cutting-plane model

$$\begin{aligned} M^k(x) &= \max_{j\in J^k} \left\{-L(p(x^j),x)\right\} \\ &= \max_{j\in J^k} \left\{\left\langle x, p_T(x^j)+p_H(x^j)-d\right\rangle - C_T(p_T(x^j)) - C_H(p_H(x^j))\right\} . \end{aligned} \tag{18}$$

As long as the relations in (15) hold, a variety of other model functions can be used, including nonpolyhedral ones (conditions that are even weaker than (15) and still maintain convergence can be found in [16]). We now explain how to exploit this versatility for the specific case under consideration. In particular, this feature allows us to build separable approximations for the nonseparable augmented Lagrangians.

5.1. Bundle Update Subproblems. To ease the reading, we use the shorten notation p_T^j for the vectors $p_T(x^j)$, and likewise for p_H. Then the expression (18) gives in (13)

$$\left\{\begin{array}{ll} \min & a+\frac{1}{2t_k}\|x-\hat{x}^k\|^2 \\ \text{s.t.} & (x,a)\in \Re^T\times\Re \\ & a\geq \left\langle x, p_T^j+p_H^j-d\right\rangle - C_T(p_T^j) - C_H(p_H^j) \text{ for } j\in J^k , \end{array}\right.$$

Following the proof of [2, Lem.10.8], it is possible to compute the dual of this strongly convex quadratic programming problem, which has the form

$$\left\{\begin{array}{ll} \min & \displaystyle\sum_{j\in J^k} \alpha_j (C_T(p_T^j)+C_H(p_H^j)) + \left\langle \hat{x}^k, d-q_T-q_H\right\rangle + \frac{t_k}{2}\|d-q_T-q_H\|^2 \\ \text{s.t.} & 0\leq \alpha_j \text{ for } j\in J^k , \displaystyle\sum_{j\in J^k}\alpha_j = 1 \\ & \text{where } q_T=\displaystyle\sum_{j\in J^k}\alpha_j p_T^j \text{ and } q_H=\sum_{j\in J^k}\alpha_j p_H^j . \end{array}\right. \tag{19}$$

Like in (13)-(14), the new price x^{k+1} can be computed using the the aggregate gradient $\hat{g}^k=\displaystyle\sum_{j\in J^k}\alpha_j^k(p_T^j+p_H^j-d)$, where the simplicial multipliers α_j^k solve (19).

To see how the dual problem (19) opens the way to model definitions that go beyond the usual cutting-plane model, suppose for the moment the costs C_T and

C_H are linear. Then (19) is equivalent to

$$\begin{cases} \min \sum_{j\in J^k} \alpha_j C_T(p_T^j) + \sum_{j\in J^k} \alpha_j C_H(p_H^j) + \langle \hat{x}^k, d - q_T - q_H \rangle + \dfrac{t_k}{2} \|d - q_T - q_H\|^2 \\ \text{s.t.} \quad q_T \in \mathcal{P}_T^k := \text{conv}\{p_T^j : j \in J^k\} \text{ and } q_H \in \mathcal{P}_H^k := \text{conv}\{p_H^j : j \in J^k\}\,. \end{cases} \tag{20}$$

With respect to problem (1), we note that

- the feasible set $\mathcal{P} = \mathcal{P}_T \times \mathcal{P}_H$ is replaced by the smaller set $\mathcal{P}_T^k \times \mathcal{P}_H^k$, and
- the objective function is an augmented Lagrangian with augmentation parameter t_k and multiplier fixed to $\hat{x}^k$.

For nonlinear generation costs, the equivalence with (19) is lost, but (20) still can be interpreted as an augmented Lagrangian of an approximation to (3). The descent rule to declare x^{k+1} a serious iterate in fact checks if that approximation is sufficiently good. The bundle update of the multiplier x^{k+1} can then be thought of as a variant of the well-known multiplier method, [8] and [20], adapted to such approximations.

The purpose of solving (13) or its variant (19) is to compute the next iterate x^{k+1}. For the example in Section 4.2 an alternative subproblem could be

$$\begin{cases} \min & C_T(q_T) + C_H(p_H) + \langle \hat{x}^k, d - q_T - p_H \rangle + \frac{t_k}{2} \|d - q_T - p_H\|^2 \\ \text{s.t.} & q_T \in \mathcal{P}_T^k\,, p_H \in \mathcal{P}_H\,. \end{cases} \tag{21}$$

As long as $C_H(\cdot)$ is quadratic and $\mathcal{P}_H$ is a polyhedron, (21) is also a quadratic program. So from the optimization point of view, the subproblem difficulty remains comparable to (19). In terms of the model (going back from (21) to a problem on x variables, like (13)), this amounts to using

$$M^k(x) = -\langle x, d\rangle + \max_{p_H \in \mathcal{P}_H} \{\langle x, p_H\rangle - C_T(p_H))\} + \max_{q_T \in \mathcal{P}_T^k} \{\langle x, q_T\rangle - C_T(q_T)\}\,,$$

a better model than (18) (in the latter, the maximum of the p_H-term is also taken over q_H in the convex hull $\mathcal{P}_H^k$).

With (21), the price update remains as in (14), for the aggregate gradient

$$\hat{g}^{k+1} = q_T^{k+1} + p_H^{k+1} - d\,, \text{ where } q_T^{k+1} \text{ and } p_H^{k+1} \text{ solve (21).}$$

As the proximal Lagrangian may need to drive r to ∞ to ensure exactness (and, hence, recover primal solutions directly from the dual solution), in view of Table 2 it may be preferable to use the Sharp Lagrangian, which amounts to replacing the rightmost objective term in (21) by $r^k \sum_{t=1}^T |d^t - q_T^t - p_H^t|$.

5.2. Tackling Nonseparability. The augmented Lagrangians (17) and (16) close the duality gap, but destroy the efficiency of the decomposition approach illustrated by Figure 1. With respect to this figure, (21) is representing the decision process on the right, that is not taken individually for each power plant, but in a block, for both the hydraulic and thermal plants. This can be potentially difficult; to make decomposition appealing also in this case, one could

- either leave the block decision process, but perform the optimization only approximately, to gain speed;

- or approximate the block decision process by individual decisions of the power plants.

We describe briefly the consequences of these approximations for the bundle method defining the update (14). We refer for details to [16]; see also [22].

On-Demand Accuracy Bundle Methods. In the first case, the minimization in (21) can be stopped before reaching optimality, by sending the solver (usually a commercial package) a threshold for the desired accuracy. By construction, the approximate solutions $\tilde{q}_T^{k+1}$ and $\tilde{p}_H^{k+1}$ define a linearization that remains below the dual function. But the oracle response is *noisy*, since we only have estimates of these objects, neither the exact function value $f(x^{k+1})$ nor an exact subgradient $g(x^{k+1})$ are available. With such *lower* linearizations, the right inequality in (15) remains true, and δ^k in (12) is always nonnegative. So the bundle method needs no modification to generate a convergent sequence of serious steps. Moreover, if the accuracy threshold is driven to zero, the method will asymptotically converge to an *exact* solution. Otherwise, the solution inherits the oracle error.

A clever variation of the asymptotically exact bundle methods described above is the variant called with *on-demand accuracy*. In this method, the approximations $\tilde{q}_T^{k+1}$ and $\tilde{p}_H^{k+1}$ satisfy the accuracy threshold *only* if they provide an estimate for the function value $f(x^{k+1})$ that satisfies the descent test. As a result, the error on the serious step sequence is controlled, and by driving it to zero, the method is once more asymptotically exact (without the need of having accurate approximations for null steps).

Jacobi-like Approximations. In (21), to separate the minimization over q_T from the minimization over p_H, the nonseparable term can be approximated as below:

$$\|d - q_T - p_H\|^2 \approx \frac{1}{2}\|d - q_T - p_H^k\|^2 + \frac{1}{2}\|d - q_T^k - p_H\|^2 .$$

The approximation splits (21) in two subproblems:

$$\begin{aligned}\langle \hat{x}^k, d\rangle + &\left\{ \begin{array}{ll} \min & C_T(q_T) - \langle \hat{x}^k, q_T\rangle + \frac{t_k}{4}\left\|d - q_T - p_H^k\right\|^2 \\ \text{s.t.} & q_T \in \mathcal{P}_T^k \end{array}\right. \\ &+ \left\{ \begin{array}{ll} \min & C_H(p_H) - \langle \hat{x}^k, -p_H\rangle + \frac{t_k}{4}\left\|d - q_T^k - p_H\right\|^2 \\ \text{s.t.} & p_H \in \mathcal{P}_H , \end{array}\right.\end{aligned}$$

corresponding to the individual squares on the right of Figure 1. Contrary to the first option, there is no guarantee that the linearization built with the estimates $\tilde{q}_T^{k+1}$ and $\tilde{p}_H^{k+1}$ solving the two problems above will be of the lower type. The right inequality in (15) holds only if adding to $f(x)$ a positive constant, bounding the error made in the estimations.

The noise produced by the approximation may make the decrease δ^k in (12) negative. To detect when noise has become too cumbersome (to the extent that the descent rule cannot be applied), the bundle method incorporates certain noise attenuation procedure to suitably modify the stepsize t_k. The variant is shown to converge, within the bound for the oracle error.

Final Comments

To ease the reading, we focused our explanations on Lagrangian decomposition for optimal power management problems. Similar ideas and techniques can be applied to solve other real-life problems, as long as they exhibit some kind of decomposability feature (along time steps, along sources of commodity, etc). This is often the case in the gas and oil industry, for example.

Bundle algorithms capable of handling inaccurate information go back to [9] and [24]. The seminal work [13] gave a general theoretical framework that was revisited in [16] to incorporate the on-demand accuracy feature, among other enhancements.

We have also chosen a particular instance, called proximal bundle method, for our development. Other variants exist, and a recent one, called *doubly stabilized* [17], offers an interesting alternative for the approximations discussed in Section 5. Level bundle methods for inexact oracles are considered in [18], [15], [28], [30].

As an illustration of the impressive gains that can be obtained with these approaches in mid-term power planning and the optimal management of hydrovalleys, we refer to [6], [27], [26].

References

[1] Bahiense, L., Maculan, N., and Sagastizábal, C. The Volume Algorithm revisited. Relation with Bundle Methods. *Math. Program., Ser. A*, 94(1):41–69, 2002.

[2] Bonnans, J., Gilbert, J., Lemaréchal, C., and Sagastizábal, C. *Numerical Optimization. Theoretical and Practical Aspects.* Universitext. Springer-Verlag, Berlin, 2006. 2nd. edition, xiv+423 pp.

[3] Cheney, E. and Goldstein, A. Newton's method for convex programming and Tchebycheff approximations. *Numerische Mathematik*, 1:253–268, 1959.

[4] Daniilidis, A. and Lemaréchal, C. On a primal-proximal heuristic in discrete optimization. *Mathematical Programming*, 104:105–128, 2005.

[5] Dubost, L., Gonzalez, R., and Lemaréchal, C. A primal-proximal heuristic applied to the French unit-commitment problem. *Mathematical Programming*, 104:129–151, 2005.

[6] Emiel, G. and Sagastizábal, C. Incremental-like bundle methods with application to energy planning. *Computational Optimization and Applications*, 46:305–332, 2010.

[7] Frangioni, A. and Gentile, C. Solving nonlinear single-unit commitment problems with ramping constraints. *Operations Research*, 54(4):767 – 775, 2006.

[8] Hestenes, M. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[9] Hintermüller, M. A proximal bundle method based on approximate subgradients. *Computational Optimization and Applications*, 20:245–266, 2001.

[10] Hiriart-Urruty, J.-B. and Lemaréchal, C. *Convex Analysis and Minimization Algorithms*. Number 305-306 in Grund. der math. Wiss. Springer-Verlag, 1993. (two volumes).

[11] Izmailov, A. and Solodov, M. *Newton-Type Methods for Optimization and Variational Problems*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, 2014.

[12] Kelley, J. E. The cutting plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.

[13] Kiwiel, K. A proximal bundle method with approximate subgradient linearizations. *SIAM J. Optim.*, 16:1007–1023, 2006.

[14] Lemaréchal, C. Lagrangian relaxation. In Jünger, M. and Naddef, D., editors, *Computational Combinatorial Optimization*, pages 112–156. Springer Verlag, 2001.

[15] Oliveira, W. and Sagastizábal, C. Level Bundle Methods for Oracles with On-Demand Accuracy. *Optimization Methods and Software*, 29(6):1180–1209, 2014.

[16] Oliveira, W., Sagastizábal, C., and Lemaréchal, C. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Mathematical Programming*, 148(1-2):241–277, 2014.

[17] Oliveira, W. and Solodov, M. A doubly stabilized bundle method for nonsmooth convex optimization. *Accepted for publication in Mathematical Programming*, 2014.

[18] Oliveira, W. L., Sagastizábal, C., and Scheimberg, S. Inexact Bundle Methods for Two-Stage Stochastic Programming. *SIAM Journal on Optimization*, 21:517–544, 2011.

[19] Pereira, M. and Pinto, L. Multi-stage stochastic optimization applied to energy planning. *Math. Program.*, 52(2):359–375, 1991.

[20] Rockafellar, R. The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and Applications*, 6:555–562, 1973.

[21] Rockafellar, R. and Wets, R.-B. *Variational Analysis*. Number 317 in Grund. der math. Wiss. Springer-Verlag, 1998.

[22] Sagastizábal, C. Divide to conquer: decomposition methods for energy optimization. *Math. Program.*, 134:187–222, 2012.

[23] Shor, N. *Minimization methods for non-differentiable functions*. Springer-Verlag, Berlin, 1985.

[24] Solodov, M. On approximations with finite precision in bundle methods for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 119(1):151–165, October 2003.

[25] Tahanan, M., van Ackooij, W., Frangioni, A., and Lacalandra, F. Large-scale unit commitment under uncertainty. *4OR*, pages 1–57, 2015.

[26] van Ackooij, W. and de Oliveira, W. Level bundle methods for constrained convex optimization with various oracles. *Computation Optimization and Applications*, 57(3):555–597, 2014.

[27] van Ackooij, W. and Sagastizábal, C. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM Journal on Optimization*, 24(2):733–765, 2014.

[28] Wolf, C., Fábián, C. I., Koberstein, A., and Stuhl, L. Applying oracles of on-demand accuracy in two-stage stochastic programming a computational study. *European Journal of Operational Research*, 239(2):437–448, 2014.

[29] Zaourar, S. and Malick, J. Prices stabilization for inexact unit-commitment problems. *Mathematical Methods of Operations Research*, 78(3):341–359, 2013.

[30] Zverovich, V., Fábián, C. I., Ellison, E. F. D., and Mitra, G. A computational study of a solver system for processing two-stage stochastic LPs with enhanced Benders decomposition. *Mathematical Programming Computation*, 4(3):211–238, 2012.

Visiting Researcher at IMPA, Estrada dona Castorina 110, 22460-310 Jardim Botânico, Brazil

E-mail: sagastiz@impa.br