Bundle relaxation and primal recovery in Unit Commitment problems. The Brazilian case.*

A. Belloni dos Santos Nogueira
IMPA, Graduate School.

A.L. Diniz Souto Lima
CEPEL - Centro de Pesquisas de Energia Elétrica, C.P. 68007, RJ 21944-970, Rio de Janeiro, Brazil.

M.E. Piñeiro Maceira
CEPEL - Centro de Pesquisas de Energia Elétrica, C.P. 68007, RJ 21944-970, Rio de Janeiro, Brazil.

C.A. Sagastizábal†
IMPA - Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, RJ 22460-320, Rio de Janeiro, Brazil. On leave from INRIA, France.

June 11, 2002

Abstract. We consider the inclusion of commitment of thermal generation units in the optimal management of the Brazilian power system. By means of Lagrangian relaxation we decompose the problem and obtain a nondifferentiable dual function that is separable. We solve the dual problem with a bundle method. Our purpose is twofold: first, bundle methods are the methods of choice in nonsmooth optimization when it comes to solve large-scale problems with high precision. Second, they give good starting points for recovering primal solutions. We use an inexact augmented Lagrangian technique to find a near-optimal primal feasible solution. We assess our approach with numerical results.

Keywords: power system operation; unit commitment; Lagrangian relaxation; bundle methods; augmented Lagrangian; feasible dispatch.

1. Introduction

The optimal scheduling of many power units is a challenging problem, from both economical and mathematical points of view. In particular, to solve the short-term unit-commitment problem (UC) for the Brazilian case means to consider simultaneously:

- thermal power-plants with high start-up costs and discontinuous operation domains, introducing non-convex constraints;
- hydro-valleys of interconnected reservoirs, leading to tight dynamic constraints (for example, water travel times between reservoirs going up to 15 days);
- hydraulic plants with nonlinear production functions depending on the water head and outflows through turbines and spillways;
- highly stochastic water inflows, linking up to 6 consecutive months, [18];
- various types of integral and logical constraints.

* Research partially supported by CEPEL.
† Corresponding author, e-mail: sagastiz@impa.br.

The corresponding optimization problem is large-scale nonconvex with mixed integer variables. For this reason, Lagrangian relaxation (or price decomposition) has become one of the most commonly used strategies in this field, see [20], [9]; although other methods were proposed in the literature, see for instance [23].

An important feature of the Lagrangian relaxation approach is that it provides marginal prices that are useful for the operators of the system. Essentially, the original primal problem is replaced by a (decomposable) dual nonsmooth optimization problem whose solution gives marginal prices. Therefore, it is of outmost importance to solve the nonsmooth dual problem with a robust method capable to attain a high degree of precision. Bundle methods are at the moment the most efficient and promising methods for nonsmooth optimization. In addition, they provide a good starting point for recovering a primal solution (i.e., a feasible dispatch) using some heuristic method, see for example [25], [11].

We consider the short-term unit-commitment problem of optimal power management in the Brazilian system. Although prior to our work there was a short-term model available [19], this model did not specify much thermal production units and considered only continuous variables. Since about 90% of the Brazilian mix is hydraulic, in a centralized management it was not important to enter into further details for thermal production. However, with the recent opening of the electric power market in Brazil, it became crucial to reflect the influence of thermal plants in marginal prices. This made it necessary to consider new UC models, with detailed operating constraints for the thermal units.

By artificially duplicating some variables, we decompose the UC problem by means of Lagrangian relaxation. Specifically, we derive a dual problem that is decomposable, .i.e., such that the dual function can be computed by solving separately two optimization problems, that we call *local problems*. The dualization scheme is chosen to take advantage of the structure of the continuous model previously available. As a result, the previous continuous model, a linear program, becomes a (hydro-thermal) local problem for our dual. The other local problem, purely thermal, is an optimization problem in a graph solved with dynamic programming techniques.

To solve the dual problem, we use the bundle method in [17]. Finally, since a solution of the dual does not solve the primal, in order to obtain primal feasible solutions, it is necessary to do a feasibility search. Starting from a primal point associated to the dual solution, we use an augmented Lagrangian technique to find a near-optimal primal feasible solution. Augmented Lagrangians are primal feasible methods that yield differentiable dual functions. We preserve separability by linearizing quadratic terms, i.e., by using an inexact method to solve the augmented dual. The procedure is independent of the data constituting the problem, and is more rigorous and systematic than other heuristics in the literature.

This paper is organized as follows. In Section 2 we explain the UC problem for the Brazilian case, including the particular modeling for power plants. The dualization scheme and the decomposable dual problem are described in Section 3. Section 4 is devoted to the bundle method used to update prices and to solve the nondifferentiable dual problem. We go back to primal considerations in Section 5, where we use another decomposition scheme combined with an inexact augmented Lagrangian method to find a primal feasible solution that is nearly optimal. Finally, in Section 6 we assess our approach by giving numerical results on test problems from the Brazilian power system.

2. General formulation

A generic formulation for the optimal power management problem is the following:

$$\begin{cases} \min\limits_{p} \mathcal{C}(p) \\ p \in \mathcal{S} \cap \mathcal{D}\,, \end{cases} \tag{1}$$

where p is the production, $\mathcal{C}$ is the operating cost function and $\mathcal{S}$, $\mathcal{D}$ represent, respectively, the static and dynamic constraints. The main elements characterizing the UC problem are given below.

2.1. Variables

We study a power mix of hydraulic valleys and thermal plants for a discrete period of time $\{1, 2, \ldots, \mathcal{T}\}$ (typically, 48 half-hour steps).

Hydraulic valleys are composed by plants and reservoirs, and each thermal plant may have several generators; we refer to each one of such components as a *unit*. We denote by p_i^t the production level of unit i at time $t \le \mathcal{T}$. Letting $\mathcal{N}$ be the total number of units in the system, $p^t := (p_1^t, p_2^t, \ldots, p_{\mathcal{N}}^t)$ stands for the production vector of whole mix at each time t; likewise, $p_i := (p_i^1, p_i^2, ..., p_i^{\mathcal{T}})$ will be the production vector of unit i for the whole horizon of time.

Sometimes we will distinguish thermal and hydraulic components and write $p_{\mathtt{T}}$ and $p_{\mathtt{H}}$.

2.2. Constraints

Static constraints are organized in two different subsets

$$\mathcal{S} = \mathcal{S}_1 \cap \mathcal{S}_2\,.$$

The first category gathers *coupling* constraints which relate production of different units (and/or plants), like satisfaction of demand, network security, spinning reserve, at each time step. They are generally represented by a large-scale linear system.

Since the previously available continuous model [19] takes into account satisfaction of demand, we find it convenient to write $\mathcal{S}_1$ as follows:

$$\mathcal{S}_1 = \mathcal{S}_{\mathtt{D1}} \cap (\mathcal{S}_{\mathtt{T1}} \times \mathcal{S}_{\mathtt{H1}})\ ,$$

where $\mathcal{S}_{\mathtt{D1}}$ corresponds to the demand constraint. Essentially, if D^t is the requirement of demand at time t, then

$$\mathcal{S}_{\mathtt{D1}} := \{p : \textstyle\sum_{i \in \mathcal{N}} p_i^t = D^t\,, t = 1, 2, \ldots, \mathcal{T}\}\,.$$

More detailed variants, including shortages and energy interchanges between subsystems of the mix, can also be considered.

Transmission constraints (bounding the capacity of each arc in the network), and security constraints (ensuring the satisfaction of the demand even after the outage of one line) can also be included in $\mathcal{S}_{\mathtt{D1}}$. In the DC flow approximation model these constraints are linear, but, as opposed to demand or reserve, they are often numerous, because the network has several hundreds of nodes. With security constraints, there will be several hundred thousand linear constraints at each time step. We do not consider them explicitly in our model. Instead we introduce additional (linear) constraints in $\mathcal{S}_{\mathtt{D}1}$ when necessary along iterations, see [19].

The sets $\mathcal{S}_{\mathtt{T}1}$ and $\mathcal{S}_{\mathtt{H}1}$ correspond to spinning reserve constraints, required at each time step for each plant. These constraints link units in the same plant and they guarantee that there is enough power started up to quickly face most random events (unit outage, errors in rain forecasts leading to underestimation of water inflows, etc). We also split them into thermal and hydraulic components, $\mathcal{S}_{\mathtt{T}1}$ and $\mathcal{S}_{\mathtt{H}1}$, respectively, and write

$$p_{\mathtt{T}} \in \mathcal{S}_{\mathtt{T}1} \iff A_{\mathtt{T}} p_{\mathtt{T}} = R_{\mathtt{T}} \quad \text{and} \quad p_{\mathtt{H}} \in \mathcal{S}_{\mathtt{H}1} \iff A_{\mathtt{H}} p_{\mathtt{H}} = R_{\mathtt{H}}\,, \tag{2}$$

where $A_{\mathtt{T}}$, $A_{\mathtt{H}}$ are matrices and $R_{\mathtt{T}}$, $R_{\mathtt{H}}$ are vectors of appropriate dimensions. In (6) below we give the explicit form for the thermal spinning reserve constraint.

The second category $\mathcal{S}_2$ represents static *non-coupling* constraints, like bounds on production levels, that are required at each time step and for each unit. Again, we split this set in thermal and hydraulic components:

$$\mathcal{S}_2 = \mathcal{S}_{\mathtt{T}2} \times \mathcal{S}_{\mathtt{H}2}\,.$$

Finally, the set of dynamic constraints $\mathcal{D} = \mathcal{D}_{\mathtt{T}} \times \mathcal{D}_{\mathtt{H}}$ includes all the operating rules for each power plant, see § 2.3 below. Typically, in this set there is a coupling of variables of the same unit/plant along time steps.

Altogether, for our UC problem the constraints in (1) are

$$\mathcal{D} \cap \mathcal{S} = (\mathcal{D}_{\mathtt{T}} \times \mathcal{D}_{\mathtt{H}}) \cap \mathcal{S}_{\mathtt{D}1} \cap (\mathcal{S}_{\mathtt{T}1} \times \mathcal{S}_{\mathtt{H}1}) \cap (\mathcal{S}_{\mathtt{T}2} \times \mathcal{S}_{\mathtt{H}2})\ . \tag{3}$$

2.3. Power plants modeling

We give here more details on the particular modeling of hydraulic and thermal plants. Note that in (1) it is assumed that each power plant i can be described straightforwardly in terms of its production p_i. Actually, realistic models use state and control variables, denoted respectively by x and u, to represent p_i as a function of x and u. At first sight, this modification comes just to a change of variables in (1). However, it is important to realize that, although (1) is formally the same, the introduction of new variables can change the structure of some constraints, that may no longer be linear, or even convex, if the dependence of p on x and u is made explicit.

2.4. Hydraulic plants

The production $p_{\mathtt{H}}$ depends on the discharge of water flowing through turbines, on the water-head and sometimes also on the spillage. These are variables that can be expressed in terms of the state and control variables, $x_{\mathtt{H}}$ and $u_{\mathtt{H}}$.

More precisely, a hydro-valley is a set of interconnected plants and reservoirs with natural inflows in each reservoir. For each reservoir $i \le \mathcal{N}_{\mathtt{H}}$, we let $j \in \mathtt{UP}_i$ be the reservoirs immediately upstream (i.e., there is no other plant or reservoir in-between upstream the valley). The state variable $x_{\mathtt{H}i}$ has components $x_{\mathtt{H}i}^{\ t}$ for each time step $t \leq \mathcal{T}$. Each component $x_{\mathtt{H}i}^{\ t}$ is the vector

$$x_{\mathtt{H}i}^{\ t} := \begin{pmatrix} V_i^t \\ E_i^t \end{pmatrix} ,$$

where V_i^t is the volume of water stored in the reservoir at the beginning of the time step and E_i^t is the evaporation during the time step. Evaporation depends on the current level of reservoirs and known rates of evaporation. Sometimes the model needs to consider travel time of water between two hydro plants. In this case, $x_{\mathtt{H}i}^{\ t}$ also includes upstream discharges for $\tau \leq t$.

The control variable $u_{\mathtt{H}i}$ has components $u_{\mathtt{H}i}^{\ t}$ for each time step $t \leq \mathcal{T}$. Each component $u_{\mathtt{H}i}^{\ t}$ describes volumes to be discharged (at the reservoir itself and at all the reservoirs immediately upstream), either by flowing through turbines (Q_i^t, Q_j^t) or by being spilt (S_i^t, S_j^t):

$$u_{\mathtt{H}i}^{\ t} := \begin{pmatrix} Q_i^t \\ S_i^t \\ (Q_j^t)_{j \in \mathtt{UP}_i} \\ (S_j^t)_{j \in \mathtt{UP}_i} \end{pmatrix} .$$

We see that to represent $p_{\mathtt{H}i} := p(x_{\mathtt{H}i}, u_{\mathtt{H}i})$ we use $\mathcal{T}(2 + 2 + 2\#\mathtt{UP}_i)$ variables.

Dynamic constraints are given by the stream-flow balance equations, establishing the physical coupling of reservoirs in the hydrovalley:

$$V_i^{t+1} + Q_i^t - \sum_{j \in \mathtt{UP}_i} Q_j^t + S_i^t - \sum_{j \in \mathtt{UP}_i} S_j^t = V_i^t + A_i^t - E_i , \tag{4}$$

where A_i^t is the natural water inflow during the time step, a deterministic forecast derived from a dynamic stochastic recourse model used for the medium term model, [18], [22]. Hence,

$$\mathcal{D}_{\mathtt{H}i} := \{p_{\mathtt{H}i} = p(x_{\mathtt{H}i}, u_{\mathtt{H}i}) : x_{\mathtt{H}i}^{\ t} \text{ and } u_{\mathtt{H}i}^{\ t} \text{ satisfy (4) for all } t < \mathcal{T}\} \quad \text{and} \quad \mathcal{D}_{\mathtt{H}} := \prod_{i \leq \mathcal{N}_{\mathtt{H}}} \mathcal{D}_{\mathtt{H}i} .$$

Since (4) is a linear equation relating $x_{\mathtt{H}i}^{\ t+1}$, $x_{\mathtt{H}i}^{\ t}$ and $u_{\mathtt{H}i}^{\ t}$, the inclusion $p_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}}$ is a linear system, possibly large.

For hydraulic plants the operating cost function $\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}{}^t)$ is the so-called *future value* of water, i.e., the cost for time steps $\tau > t$ that will result from not having enough water in the reservoirs. This cost is given by the extra thermal generation and/or shortages of power that might have to be faced in the future, due to the lack of water. It is estimated by using a cutting planes approximation built with information from the medium term model [18]:

$$\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}{}^t) \approx \max_{k \leq K} \{\Pi_t^k x_{\mathtt{H}}{}^t + \delta_t^k\} ,$$

where K is the number of cutting planes and, for each $k \leq K$, Π_t^k is a matrix and δ_t^k is a vector of appropriate dimensions. Essentially, matrices Π measure the sensitivity of $\mathcal{C}_{\mathtt{H}}$ to variations on the state variables, typically the volume of water stored in the reservoirs. A distinctive feature of the future value function is that it is *not decomposable*, i.e., it cannot be expressed as a sum over the hydraulic mix of individual costs.

Finally, non coupling static constraints are given by the *production* function for the reservoirs. As we mentioned, each plant production p_i depends on the discharge of water flowing through turbines and on the water-head. In turn, water-head depends on volumes upstream and downstream the reservoir. Also, for some plants production varies with spillage. For this reason, the non convex production function is first convexified and this convex hull is then approximated by a piecewise linear function obtained using a Taylor expansion on the control variables (specifically, on Q_i^t and S_i^t):

$$\mathcal{S}_{\mathtt{H}2} : \{p_{\mathtt{H}} = p(x_{\mathtt{H}}, u) : p_i^t = \min_{k \le K_i} \{P_x^k {x_{\mathtt{H}}}_i^t + P_u^k {u_{\mathtt{H}}}_i^t + d_i^k\} \quad \text{for all } i \le \mathcal{N}_{\mathtt{H}} \text{ and } t \le \mathcal{T}\},$$

where for each $i \le \mathcal{N}_{\mathtt{H}}$, K_i is the number of pieces defining the linear function and, for each $k \le K_i$, P_x^k, P_u^k are matrices and d_i^k is a vector of appropriate dimensions.

2.5. Thermal plants

For hydraulic plants both state and control variables are continuous. This is no longer the case for thermal plants, since on/off constraints require the use of integer variables. Thermal dynamic constraints are modeled by a graph whose nodes are associated with an integer operating mode $({u_{\mathtt{T}}}_i^t)$, a production level $({x_{\mathtt{T}}}_i^t)$ and a cost $(\mathcal{C}_{\mathtt{T}})$.

A thermal unit can only be operated after completing start-up phases and following some fixed output curves for the production during shutdown or start-up times. There are also maintenance periods where the unit must be off line. Besides, not any range of variation is allowed for the production.

Therefore, for thermal units we have that $p_i^t = p({x_{\mathtt{T}}}_i^t, {u_{\mathtt{T}}}_i^t)$; in particular, when the unit is on line, ${u_{\mathtt{T}}}_i^t > 0$ and the production level satisfies ${x_{\mathtt{T}}}_i^t \in [\underline{x_{\mathtt{T}i}}, \overline{x_{\mathtt{T}i}}]$.

Suppose that $\mathcal{N}_{\mathtt{T}}$ plants compose the thermal mix, and each plant j has K_j units indexed in the set J_j. In order to model the plant operation by means of a graph for each unit i we define the set of nodes as $\mathcal{V}_{G_i} := \mathcal{V}_i \setminus \mathcal{M}_i$, where

$$\mathcal{V}_i := \{(t, {u_{\mathtt{T}}}_i^t) \in \{1, \ldots, \mathcal{T}\} \times \{0,1\}\} \text{ and } \mathcal{M}_i := \{(t,1) \in \{1, \ldots, \mathcal{T}\} \times \{0,1\} : m_i^t = 1\}.$$

The binary variables ${u_{\mathtt{T}}}_i^t$ and m_i^t are associated, respectively, to the operating and maintenance modes. At time t, we set ${u_{\mathtt{T}}}_i^t = 0$ if the i^{th} unit is off line and, likewise, $m_i^t = 1$ if the i^{th} unit is off for maintenance.

The set of edges E_{G_i} consists of elements $(v, w) \in \mathcal{V}_{G_i} \times \mathcal{V}_{G_i}$ satisfying

$$\begin{cases} v_t = w_t + 1 & \text{if } v_u = w_u \\ v_t = w_t + t_i^{up} & \text{if } v_u = w_u - 1 \\ v_t = w_t + t_i^{down} & \text{if } v_u = w_u + 1\,, \end{cases} \tag{5}$$

where t_i^{up} (resp. t_i^{down}) is the minimum time required to start up (resp. to shut down) unit i.

Finally, having the induced graph $G_i = (\mathcal{V}_{G_i}, E_{G_i})$, we denote by $\mathcal{G}_i$ those binaries sequences representing connected paths between a node with $t = 1$ and another with $t = \mathcal{T}$.

Altogether, dynamic constraints are given by

$$\mathcal{D}_{\mathtt{T}i} := \{p_{\mathtt{T}i} = p(x_{\mathtt{T}i}, u_{\mathtt{T}i}) \in \mathbb{R}^{\mathcal{T}} : u_{\mathtt{T}i} \in \mathcal{G}_i\} \quad \text{and} \quad \mathcal{D}_{\mathtt{T}} := \prod_{i \in J_j, j \le \mathcal{N}_{\mathtt{T}}} \mathcal{D}_{\mathtt{T}i}\,.$$

Brazil's legislation requires spinning reserve to be available at *each plant*, not just in the whole system. Let R_j^t be the amount of reserve required for the j^{th} plant at time t. Each unit i in the plant has $r_i(x_{\mathtt{T}}{}_i^t) := \overline{x_{\mathtt{T}i}} - x_{\mathtt{T}}{}_i^t$ power available for reserve, so we have that

$$\mathcal{S}_{\mathtt{T}1} := \left\{ p_{\mathtt{T}}{}^t = p(x_{\mathtt{T}}{}^t, u_{\mathtt{T}}{}^t) : \sum_{i \in J_j} r_i(x_{\mathtt{T}}{}_i^t) \ge R_j^t\,, \quad \text{for all } j \le \mathcal{N}_{\mathtt{T}} \right\}, \tag{6}$$

a linear expression, as in (2), coupling all the units in the same plant. In contrast, maximum and minimum generation constraints

$$\mathcal{S}_{\mathtt{T}2} := \left\{ p_{\mathtt{T}} = p(x_{\mathtt{T}}, u) : u_{\mathtt{T}}{}_i^t \underline{x_{\mathtt{T}}{}_i^t} \le x_{\mathtt{T}}{}_i^t \le u_{\mathtt{T}}{}_i^t \overline{x_{\mathtt{T}i}} \quad \text{for all } i \in J_j\,, j \le \mathcal{N}_{\mathtt{T}} \text{ and } t \le \mathcal{T} \right\},$$

are neither coupling time steps nor units.

Note that, as opposed to hydraulic plants, there is no *production* function for thermal plants: the state variable $x_{\mathtt{T}}$ gives in a straightforward manner the production level $p_{\mathtt{T}}$. In addition, the cost function $\mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}}{}^t)$ is the sum over the thermal units of individual operating costs, generally given by piecewise linear functions.

3. Decomposition scheme

We now explain the dualization scheme we use to decompose the UC problem (1) and obtain a dual function defined as the sum of two optimization problems, that we call *local problems*.

3.1. The primal and dual problems

Using the expression (3) in (1) we obtain the UC problem in its primal form:

$$\begin{cases} \min\limits_{p_{\mathtt{T}}, p_{\mathtt{H}}} \mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}}) + \mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}) \\ p_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}1} \cap \mathcal{S}_{\mathtt{T}2} \\ p_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}1} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}}, p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1}\,. \end{cases} \tag{7}$$

To achieve decomposition, we first write the (coupling) spinning reserve constraints $p_{\mathtt{T}} \in \mathcal{S}_{\mathtt{T}1}$, $p_{\mathtt{H}} \in \mathcal{S}_{\mathtt{H}1}$ using the matrix form (2):

$$\begin{cases} \min\limits_{p_{\mathtt{T}}, p_{\mathtt{H}}} \mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}}) + \mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}) \\ p_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}2} \\ p_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}}, p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1} \\ A_{\mathtt{T}} p_{\mathtt{T}} = R_{\mathtt{T}} \\ A_{\mathtt{H}} p_{\mathtt{H}} = R_{\mathtt{H}}\,. \end{cases}$$

Finally, we introduce the artificial variable $q_{\mathtt{T}}$ duplicating $p_{\mathtt{T}}$ and obtain the following problem, equivalent to (7):

$$\begin{cases} \min\limits_{p_{\mathtt{T}},p_{\mathtt{H}},q_{\mathtt{T}}} \mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}})+\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}) \\ q_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}2} \\ p_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}},p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1} \\ A_{\mathtt{T}}q_{\mathtt{T}} = R_{\mathtt{T}} \\ A_{\mathtt{H}}p_{\mathtt{H}} = R_{\mathtt{H}} \\ p_{\mathtt{T}} = q_{\mathtt{T}}\,. \end{cases}$$

We relax the last three equations. Letting $\lambda := (\lambda_A\,, \lambda_{R_{\mathtt{T}}}\,, \lambda_{R_{\mathtt{H}}})$ be the corresponding multipliers, we obtain the following problem, equivalent to (7):

$$\begin{cases} \min\limits_{p_{\mathtt{T}},p_{\mathtt{H}},q_{\mathtt{T}}} \max\limits_{\lambda} \mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}})+\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}})+\lambda_A^{\top}(p_{\mathtt{T}}-q_{\mathtt{T}})+ \\ \qquad\qquad\qquad\qquad \lambda_{R_{\mathtt{T}}}^{\top}(A_{\mathtt{T}}q_{\mathtt{T}}-R_{\mathtt{T}})+\lambda_{R_{\mathtt{H}}}^{\top}(A_{\mathtt{H}}p_{\mathtt{H}}-R_{\mathtt{H}}) \\ q_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}2} \\ p_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}},p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1}\,. \end{cases}$$

The dual problem is obtained when interchanging "max" and "min" in the problem above

$$\begin{cases} \max\limits_{\lambda} \min\limits_{p_{\mathtt{T}},p_{\mathtt{H}},q_{\mathtt{T}}} \mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}})+\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}})+\lambda_A^{\top}(p_{\mathtt{T}}-q_{\mathtt{T}})+ \\ \qquad\qquad\qquad\qquad \lambda_{R_{\mathtt{T}}}^{\top}(A_{\mathtt{T}}q_{\mathtt{T}}-R_{\mathtt{T}})+\lambda_{R_{\mathtt{H}}}^{\top}(A_{\mathtt{H}}p_{\mathtt{H}}-R_{\mathtt{H}}) \\ q_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}2} \\ p_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}},p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1}\,. \end{cases} \tag{8}$$

Decomposability is more apparent when introducing the dual functions

$$\theta_p(\lambda) := \begin{cases} \min\limits_{p_{\mathtt{T}},p_{\mathtt{H}}} \mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}})+\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}})+\lambda_A^{\top}p_{\mathtt{T}}+\lambda_{R_{\mathtt{H}}}^{\top}A_{\mathtt{H}}p_{\mathtt{H}} \\ p_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}},p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1}\,. \end{cases} \tag{9}$$

and

$$\theta_q(\lambda) := \begin{cases} \min\limits_{q_{\mathtt{T}}} -\lambda_A^{\top}q_{\mathtt{T}}+\lambda_{R_{\mathtt{T}}}^{\top}A_{\mathtt{T}}q_{\mathtt{T}} \\ q_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}2}\,. \end{cases} \tag{10}$$

so that (8) becomes

$$\begin{cases} \max\limits_{\lambda=(\lambda_A,\lambda_{R_{\mathtt{T}}},\lambda_{R_{\mathtt{H}}})} \theta_p(\lambda)+\theta_q(\lambda)-\lambda_{R_{\mathtt{T}}}^{\top}R_{\mathtt{T}}-\lambda_{R_{\mathtt{H}}}^{\top}R_{\mathtt{H}} \end{cases} \tag{11}$$

A natural question is to ask how good is the dual problem (11) in terms of the primal problem (7). Call $\bar{c}$ the optimal objective value in the primal problem (7). It is known that the dual optimal objective value $\bar{\Theta}$ is lower than $\bar{c}$, and a crucial property is whether it is equal to

$\bar{c}$. However, this has no reason to be true, because (7) is a nonconvex optimization problem. In this sense, the dual approach can only yield approximate solutions. Clearly, for such approach to be attractive, it is crucial that the (positive) *duality gap* $\bar{c} - \bar{\Theta}$ is sufficiently small. In [16] it is given an interpretation of the optimal dual solution $\bar{\lambda}$ as marginal prices associated with particular convexified forms of (7), see also [11]. A consequence of this interpretation is that if some nonconvex constraints are duplicated, then the duality gap is reduced.

Similar decomposition schemes have been proposed for nonlinear unit-commitment problems in [2], [15], [1]. They generalize an earlier approach in [21] for mixed integer linear programming.

3.2. Evaluation of the dual function

We call *local problems* the optimization problems that are needed to solve in order to compute, for each price λ, the dual objective function. For (11), there are two local problems, corresponding to the evaluation of θ_p and θ_q via (9) and (10), respectively.

The evaluation of θ_p amounts to solving a linear program with hydraulic and thermal continuous variables, similar to the problem solved prior to our work in the continuous model [19].

Evaluation of θ_q, by contrast, corresponds to an optimization problem in graphs. Since $\mathcal{C}_{\mathtt{T}} = \sum_i \mathcal{C}_{\mathtt{T}i}$ and reserve constraints have been dualized, (10) can be decomposed into units, yielding simple subproblems that can be easily solved by dynamic programming. However, when operating costs $\mathcal{C}_{\mathtt{T}i}$ are linear, a bang-bang phenomenon is observed in the solutions (i.e., they oscillate between the two extreme allowed values, $\underline{x_{\mathtt{T}i}}$ and $\overline{x_{\mathtt{T}i}}$). To avoid "bang-bang" effects, it can be useful to add a small quadratic term to thermal costs (see $\mathcal{C}_{\mathtt{T}qua}$ in Table III on page 18 below).

4. Solving the dual problem

A dual function such as θ_p or θ_q, defined by a minimization problem such as (9) or (10), is concave and so is

$$\Theta(\lambda) := \theta_p(\lambda) + \theta_q(\lambda) - \lambda_{R_{\mathtt{T}}}^\top R_{\mathtt{T}} - \lambda_{R_{\mathtt{H}}}^\top R_{\mathtt{H}}\,,$$

the objective function in (11). The evaluation of this function at a particular $\lambda = \lambda^n$ amounts to compute $\theta_p(\lambda^n)$ and $\theta_q(\lambda^n)$ in (9) and (10). Associated to such computations there are minimizers $(p_{\mathtt{T}}(\lambda^n), p_{\mathtt{H}}(\lambda^n))$ and $q_{\mathtt{T}}(\lambda^n)$, respectively. These optimal solutions give the following subgradient of the dual function:

$$\partial\Theta(\lambda^n) \ni g(\lambda^n) := (p_{\mathtt{T}}(\lambda^n) - q_{\mathtt{T}}(\lambda^n), A_{\mathtt{T}}q_{\mathtt{T}}(\lambda^n) - R_{\mathtt{T}}, A_{\mathtt{H}}p_{\mathtt{H}}(\lambda^n) - R_{\mathtt{H}})\,.$$

We now analyze different methods that can be applied to solve (11).

4.1. Nonsmooth optimization algorithms. General considerations

As a general rule, and for the sake of consistency, nonsmooth optimization (NSO) methods strive to mimic as much as possible those of differentiable optimization. Thus, at first glance, analyzing separately the nonsmooth case could seem useless. Yet, NSO has some traps that call for special approaches. For example, the issue of implementing reliable stopping tests is extremely delicate, because the condition "$g \in \partial\Theta(\lambda)$ with $\|g\| \leq \varepsilon$", directly translated from "$\|\nabla\Theta(\lambda)\| \leq \varepsilon$", may never happen. This situation occurs even in very simple cases: the absolute value function $\Theta(\lambda) = -|\lambda|$, defined for $\lambda \in \mathbb{R}$, has $|g| = 1$ for all $\lambda \neq 0$, the optimum point.

Another important consideration is *stability* of the method. Both subgradient and cutting-planes methods for NSO are unstable. Depending on the particular context, these algorithms either may stall and fail to converge, or they may converge to a nonoptimal point.

When **high accuracy** is required, it is crucial to apply an algorithm that is numerically stable and has a good stopping test. Bundle methods are well-known for their robustness, reliability and precision.

We now address these issues more extensively for three NSO methods, namely subgradient, cutting-planes and bundle methods. For full details we refer to [5, Part II].

4.2. Subgradient and Cutting-Planes methods

Consider the maximization of Θ, with the help of local solvers able to compute $\Theta(\lambda)$, yielding $g(\lambda) \in \partial\Theta(\lambda)$.

Subgradient methods, see [24], use the information given by the local solvers only once at a time, without a memory of past iterations. More specifically, given the current λ^n, the next iterate is defined as

$$\lambda^{n+1} := \lambda^n + t_n \frac{g(\lambda^n)}{\|g(\lambda^n)\|}. \tag{12}$$

A major difficulty of these methods is to define practical rules for the stepsize t_n which ensure convergence. A common choice is $t_n = m\frac{\bar{\Theta} - \Theta(\lambda^n)}{\|g(\lambda^n)\|}$, where $m \in]0, 2[$ is a parameter and $\bar{\Theta}$ is the optimal value in (11). If $\bar{\Theta}$ is not known, one is forced to use estimates, As a result, the algorithm may fail to converge, or if it converges, the limit-point need not be optimal, see [12].

Subgradient methods suffer from important drawbacks, such as lack of implementable stopping test, convergence to non-optimal points, possible poor rate of convergence. Nevertheless, they are extremely popular among practitioners, because of their simplicity of implementation. A crucial issue is that, since subgradient methods have to be stopped using heuristics (either the number of iterations becomes too big, or $\Theta(\lambda^{n+1}) - \Theta(\lambda^n)$ and/or $\lambda^{n+1} - \lambda^n$ become too small), they are not suitable for problems where high accuracy is required. Incidentally, this feature in confirmed by our numerical results comparing subgradient and bundle methods, see Section 6 below.

Cutting-planes and bundle methods overcome some of these drawbacks. The price to pay will be an increasing complexity when updating iterates.

In particular, keeping track of past information allows to define a model of the objective function. Cutting-planes methods, [7], [13], use the past values given by the local solvers, i.e., $\{\Theta(\lambda^k), g(\lambda^k)\}_{k \leq n}$, to construct a piecewise-linear approximation of Θ:

$$\hat{\Theta}_n(\lambda) := \min_{k \leq n}[\Theta(\lambda^k) + g(\lambda_k)^\top(\lambda - \lambda^k)].$$

The new iterate is found by maximizing the model on a convex compact set S to be determined apriori: λ^{n+1} solves

$$\begin{cases} \max\limits_{\lambda \in S, r \in \mathbb{R}} r \\ r \leq \Theta(\lambda^k) + g(\lambda_k)^\top(\lambda - \lambda^k) \text{ for } k \leq n. \end{cases} \tag{13}$$

When S is polyhedral, this is a linear programming problem.

Note that by definition of $\hat{\Theta}_n$ and the concavity of Θ, the graph of the cutting-planes model approaches the graph of Θ from inside, with increasing accuracy as n grows. This is why the method is convergent. In addition, there is now an implementable stopping test given by the positive quantity $\delta_n := \hat{\Theta}_n(\lambda^{n+1}) - \Theta(\lambda^{n+1})$m called *nominal increase*. When δ_n is smaller than a tolerance `tol`, the algorithm stops with a point λ^{n+1} satisfying $\Theta(\lambda^{n+1}) \geq \Theta(\lambda) - \texttt{tol}$ for all $\lambda \in S$.

Like in subgradient methods, cutting-planes methods do not necessarily generate a sequence of increasing objective values $\Theta(\lambda^n)$. Basically, the closer to an optimum is λ^n, the worse will be the next iterate. This is due to the introduction in the model $\hat{\Theta}_n$ of an almost "horizontal" affine function $\Theta(\lambda^n)+g(\lambda^n)^\top(\lambda-\lambda^n)$. As a result, the minimizer λ^{n+1} of the model is "pushed away" from the minimum of Θ. When "horizontal" affine functions are added to the model, iterates start oscillating and the algorithm may present instabilities and bad numerical behavior. Bundle methods propose variants to force $\Theta(\lambda^n)$ to increase, while keeping the model idea, important to define stopping tests.

Another serious drawback of cutting-planes methods is the indefinite accumulation of affine functions defining the model. This phenomenon is amplified by the instability problem mentioned above. More precisely, as n grows, the linear program defining iterates has more and more constraints. Because of instability, many constraints are similar. So the linear program becomes extremely difficult to solve. Not only due to its size, but also due to its bad conditioning. A way out of this tailing-off effect is to *clean the model*, and eliminate the *less active* constraints in the linear program, following some criteria. By the nature of cutting-planes methods, the selection of "inactive" constraints can only be done using heuristics, which are not supported by any convergence result. Bundle methods, by contrast, use the so-called *aggregation* technique to control the size (and, possibly, conditioning) of problems defining iterates *without impairing* the original properties of global convergence.

4.3. Bundle methods

None of the previous methods can guarantee an increase of the objective function at each iteration. We now consider bundle methods, which have this remarkable ascent property, as well as stability.

In order to avoid oscillations *à la* cutting-planes, it is important to "remember" the best point obtained. Bundle methods collect relevant data along iterations in the following *bundle* of information:

$$\{\Theta(\lambda^k), g(\lambda^k)\}_{k \leq \min(n, \text{MAXBUN})} \quad \text{and } \hat{\lambda}, \text{ the point with biggest objective value,}$$

where MAXBUN is a positive integer limiting the maximum size of the bundle. Thanks to this additional information, the algorithm can generate two sequences of points. One is the sequence of sample points used to define the model $\hat{\Theta}_n$. We call these points *candidate* points and denote them by λ^c. A second sequence consists of those sample points where the objective function increases. We call these points *stability* or *prox-centers* and denote them by $\hat{\lambda}$. Note that $\{\hat{\lambda}\}$ is a subsequence of $\{\lambda^c\}$.

Instead of the linear program (13), candidate points are found by solving quadratic programming problems of the form

$$\max_{\lambda} [\hat{\Theta}_n(\lambda) - \frac{1}{2}\mu\|\lambda - \hat{\lambda}\|^2], \tag{14}$$

where $\mu > 0$ is a parameter playing the role of a line-search stepsize in this nonlinear context. Candidate points are declared new stability centers only when they provide a sufficient sscent for Θ:

$$\Theta(\lambda^c) \geq \Theta(\hat{\lambda}) + m\delta^c$$

where $m \in]0,1[$ is a parameter and $\delta^c = \hat{\Theta}(\lambda^c) - \Theta(\hat{\lambda})$ measures the increase predicted by the model. By the optimality condition of (14), there exists a certain $\varepsilon \geq 0$ and a certain "regularized subgradient" G, which is an ε-subgradient of Θ at the prox-center $\hat{\lambda}$ such that

$$\lambda^c = \hat{\lambda} + \frac{1}{\mu}G \quad \text{and} \quad \delta^c = \varepsilon + \frac{1}{\mu}\|G\|^2. \tag{15}$$

As a result, the cutting-planes stoppping test $\delta^c \leq \mathtt{tol}$ can be split. We stop when both ε and $\|G\|$ are small, i.e., when $\varepsilon \leq \mathtt{tol}_\varepsilon$ and $\|G\| \leq \mathtt{tol}_G$, for given tolerances $\mathtt{tol}_\varepsilon$ and $\mathtt{tol}_G$. Note that in this case, denoting by $\bar{\lambda}$ a maximizer of Θ, we can write

$$\Theta(\bar{\lambda}) \leq \Theta(\hat{\lambda}) + G^\top(\bar{\lambda} - \hat{\lambda}) + \varepsilon \leq \Theta(\hat{\lambda}) + \|G\|\,\|\bar{\lambda} - \hat{\lambda}\| + \varepsilon\,.$$

Hence, bundle methods stop with an almost optimal solution, $\hat{\lambda}$.

It should be noted that (14) is a convex quadratic programming problem (QP). Cutting-planes methods define iterates by solving linear programs (LP). This additional complication is the price to pay to obtain stability. Nevertheless, and contrary to a widespread belief, convex QPs are not much more difficult to solve than LPs. The underlying resolution technique is essentially the same (active sets strategies, interior points, etc). In actual implementations a QP such as (14) is solved through its dual. This dual QP has a very special structure, namely, a simplicial feasible set. Good QP solvers exploit such structure, as well as the fact of knowing how QP-data changes along iterations in order to define warm starts. As a result, solving (14) does not require much more computational work than solving an LP. Finally, a pay-off compensating the extra computational burden is that having QP problems allows to compress the bundle of information (i.e., to *clean the model*) by using the so-called aggregation techniques. Aggregation is the synthesis mechanism that condenses the essential information of the bundle into one single couple, given by (ε, G), see [5, Part II]. With this technique, the size of (14) is kept controlled along iterations (as opposed to cutting-planes methods), without affecting the good convergence properties of the method.

For our implementation, we apply a penalized variant of bundle methods, called "poor-man" and introduced in [17]. We refer to this work for a complete description of each step of the algorithm, which essentially follows the main lines given here.

5. Back to the primal

To recover primal solutions we use an inexact augmented Lagrangian technique, similar to the ones in [4], [10], [8].

5.1. The augmented dual problem

We now explain the decomposition scheme chosen. Starting from (7), we introduce artificial variables both for thermal and hydraulic production, $q_{\mathtt{T}}$ and $q_{\mathtt{H}}$, respectively, and write the primal problem in the equivalent form

$$\begin{cases} \min\limits_{p_{\mathtt{T}},p_{\mathtt{H}},q_{\mathtt{T}},q_{\mathtt{H}}} \frac{1}{2}\Big(\mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}}) + \mathcal{C}_{\mathtt{T}}(q_{\mathtt{T}})\Big) + \frac{1}{2}\Big(\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}) + \frac{1}{2}\mathcal{C}_{\mathtt{H}}(q_{\mathtt{H}})\Big) \\ q_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}1} \cap \mathcal{S}_{\mathtt{T}2} \\ q_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}1} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}}, p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1} \\ p_{\mathtt{T}} = q_{\mathtt{T}} \\ p_{\mathtt{H}} = q_{\mathtt{H}} \,. \end{cases}$$

We relax the last two equations with multipliers $\lambda_{A_{\mathtt{T}}}$ and $\lambda_{A_{\mathtt{H}}}$ and use the augmented Lagrangian

$$\begin{aligned} L_c(p, q, \lambda_{A_{\mathtt{T}}}, \lambda_{A_{\mathtt{H}}}) \;:=\; & \tfrac{1}{2}\Big(\mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}}) + \mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}) + \mathcal{C}_{\mathtt{T}}(q_{\mathtt{T}}) + \mathcal{C}_{\mathtt{H}}(q_{\mathtt{H}})\Big) + \lambda_{A_{\mathtt{T}}}^\top(p_{\mathtt{T}} - q_{\mathtt{T}}) + \lambda_{A_{\mathtt{H}}}^\top(p_{\mathtt{H}} - q_{\mathtt{H}}) \\ & + \tfrac{c}{2}\|p_{\mathtt{T}} - q_{\mathtt{T}}\|^2 + \tfrac{c}{2}\|p_{\mathtt{H}} - q_{\mathtt{H}}\|^2 \,, \end{aligned}$$

where c is a positive parameter. Due to the introduction of the quadratic term, augmented Lagrangians yield differentiable dual functions. Furthermore, a dual solution gives a primal feasible point.

The procedure will be to increase c along iterations in order to force satisfaction of the relaxed constraints, and, at the same time, maximize the associated dual function

$$\Theta_c(\lambda) := \begin{cases} \min\limits_{p=(p_{\mathtt{T}},p_{\mathtt{H}}),\, q=(q_{\mathtt{T}},q_{\mathtt{H}})} L_c(p, q, \lambda) \\ q_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}1} \cap \mathcal{S}_{\mathtt{T}2} \\ q_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}1} \cap \mathcal{S}_{\mathtt{H}2} \\ (p_{\mathtt{T}}, p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1} \,, \end{cases} \tag{16}$$

with an inexact gradient-like method, see [3, Ch. 4] and references therein. In particular, in order to avoid ill conditioning, the rate of increase of the penalty parameter c must be relatively small and a tradeoff is required.

The algorithm needs to compute at each iteration, say for a given λ^n, the values $\Theta_c(\lambda^n)$ and $\nabla\Theta_c(\lambda^n) = p^n - q^n$, where $p^n := p(\lambda^n)$ and $q^n := q(\lambda^n)$ solve (16) when $\lambda = \lambda^n$. Note, however, that the same quadratic term yielding differentiability prevents the dual problem from being decomposable. A possibility to force decomposability is to somehow linearize the quadratic terms linking $p_{\mathtt{T}}$ with $q_{\mathtt{T}}$ and $p_{\mathtt{H}}$ with $q_{\mathtt{H}}$.

We proceed as follows. Suppose at iteration n we have computed values ${p_{\mathtt{T}}}^n$, ${q_{\mathtt{T}}}^n$, ${p_{\mathtt{H}}}^n$ and ${q_{\mathtt{H}}}^n$. Then we approximate

$$\frac{1}{2}c\|p_{\mathtt{T}} - q_{\mathtt{T}}\|^2 \approx \frac{1}{2}c\|p_{\mathtt{T}} - \frac{{p_{\mathtt{T}}}^n + {q_{\mathtt{T}}}^n}{2}\|^2 + \frac{1}{2}c\|\frac{{p_{\mathtt{T}}}^n + {q_{\mathtt{T}}}^n}{2} - q_{\mathtt{T}}\|^2$$

and likewise for hydraulic variables, mutatis mutandis.

With this approximation, the minimization in (16) is replaced by the sum of three uncoupled (quadratic) problems,

$$
\begin{aligned}
\Theta_c(\lambda) \;\approx\; & \left\{ \begin{array}{l} \min\limits_{p_{\mathtt{T}},p_{\mathtt{H}}} \frac{1}{2}\mathcal{C}_{\mathtt{T}}(p_{\mathtt{T}}) + \frac{1}{2}\mathcal{C}_{\mathtt{H}}(p_{\mathtt{H}}) + \lambda_{A_{\mathtt{T}}}^{\top} p_{\mathtt{T}} + \lambda_{A_{\mathtt{H}}}^{\top} p_{\mathtt{H}} \\ \qquad\qquad\qquad + \frac{c}{2}\|p_{\mathtt{T}} - \frac{{p_{\mathtt{T}}}^n + {q_{\mathtt{T}}}^n}{2}\|^2 + \frac{c}{2}\|p_{\mathtt{H}} - \frac{{p_{\mathtt{H}}}^n + {q_{\mathtt{H}}}^n}{2}\|^2 \\ (p_{\mathtt{T}}, p_{\mathtt{H}}) \in \mathcal{S}_{\mathtt{D}1} \end{array} \right. \\
& + \left\{ \begin{array}{l} \min\limits_{q_{\mathtt{T}}} \frac{1}{2}\mathcal{C}_{\mathtt{T}}(q_{\mathtt{T}}) - \lambda_{A_{\mathtt{T}}}^{\top} q_{\mathtt{T}} + \frac{c}{2}\|\frac{{p_{\mathtt{T}}}^n + {q_{\mathtt{T}}}^n}{2} - q_{\mathtt{T}}\|^2 \\ q_{\mathtt{T}} \in \mathcal{D}_{\mathtt{T}} \cap \mathcal{S}_{\mathtt{T}1} \cap \mathcal{S}_{\mathtt{T}2} \end{array} \right. \\
& + \left\{ \begin{array}{l} \min\limits_{q_{\mathtt{H}}} \frac{1}{2}\mathcal{C}_{\mathtt{H}}(q_{\mathtt{H}}) - \lambda_{A_{\mathtt{H}}}^{\top} q_{\mathtt{H}} + \frac{c}{2}\|\frac{{p_{\mathtt{H}}}^n + {q_{\mathtt{H}}}^n}{2} - q_{\mathtt{H}}\|^2 \\ q_{\mathtt{H}} \in \mathcal{D}_{\mathtt{H}} \cap \mathcal{S}_{\mathtt{H}1} \cap \mathcal{S}_{\mathtt{H}2}\,. \end{array} \right.
\end{aligned}
\tag{17}
$$

Since the linearization of the quadratic term is valid only locally, it is fundamental to have a good starting point available. These points are provided by the bundle method, together with a solution of (11). More precisely, if $\hat{\lambda}$ is the numerical solution of (11) found by the bundle method, then the minimizers $(p_{\mathtt{T}}(\hat{\lambda}), p_{\mathtt{H}}(\hat{\lambda}))$ and $q_{\mathtt{T}}(\hat{\lambda})$ in (9) and (10) written for $\lambda = \hat{\lambda}$ are good starting points.

5.2. Evaluation of the augmented dual function

In the approximation (17) the evaluation of Θ_c is done by solving three local problems:

- a local hydro-termal problem with continuous variables and demand constraints;
- a local problem purely thermal, with integer variables (recall the definition of $\mathcal{D}_{\mathtt{T}}$) and reserve constraints;
- a local problem purely hydraulic, with continuous variables and reserve constraints.

All the local problems have quadratic objective function. We solve the two continuous quadratic local problems with the package PLCBAS, based on the method in [6], specially adapted to ill conditioned problems.

The purely thermal local problem with integer variables is again an optimization problem in a graph, coupled by units but uncoupled by plants.

To solve this problem, suppose first that there are no start-up and shutdown constraints in (5). We create a new graph generated by the units of the same plant where we take into account reserve constraints. Letting $K_j := \#J_j$ be the number of units in the j^{th} plant, we define binary sequences u of length K_j to represent which units in the plant are on/off. In particular, denoting by $\delta(u,i) := u \bmod 2^i$, the expression $\delta(u,i) = 1$ means that the i^{th} unit is on line at the configuration u.

Altogether, we define the nodes $\mathcal{V}_{G_j} := \hat{\mathcal{V}}_j \setminus (\cup_{i\in J_j}\hat{\mathcal{M}}_i \cup O_j)$, where

$$\hat{\mathcal{V}}_i := \{(t,u) \in \{1,\ldots,\mathcal{T}\} \times \{0,1\}^{K_j}\}\,,$$

$$\hat{\mathcal{M}}_i := \{(t,u) \in \{1,\ldots,\mathcal{T}\} \times \{0,1\}^{K_j} : \delta(u,i) = 1, m_i^t = 1\},$$

and

$$\hat{\mathcal{O}}_j := \{(t,u) \in \{1,\ldots,\mathcal{T}\} \times \{0,1\}^{K_j} : \sum_{j:\delta(u,i)=1} (\overline{x_{\mathtt{T}i}} - \underline{x_{\mathtt{T}i}}) < R_j^t\}.$$

The set $\hat{\mathcal{O}}_j$ contains those configurations for which the spinning reserve constraint is not attainable. Barring the start-up and shut-down relations in (5) yields the edges

$$E_{G_j} := \{(v,w) \in \mathcal{V}_{G_j} \times \mathcal{V}_{G_j} : v_t = w_t + 1\}.$$

To associate costs to arcs incident to a node $(t, {u_\mathtt{T}}^t)$, we solve a small convex problem with K_j variables:

$$\begin{cases} \min\limits_{{x_\mathtt{T}}_j^t := ({x_\mathtt{T}}_i^t)_{i\in J_j}} & {\mathcal{C}_\mathtt{T}}^t({x_\mathtt{T}}_j^t) \\ {x_\mathtt{T}}_j^t \in {\mathcal{S}_\mathtt{T}}_{1_j}^t \cap {\mathcal{S}_\mathtt{T}}_{2_j}^t & \\ {x_\mathtt{T}}_i^t = 0 \text{ if } \delta({u_\mathtt{T}}^t, i) = 0 & \text{for each } i \in J_j\,. \end{cases}$$

We are then in a position to apply again a standard dynamic programming methodology. While the complexity of resolution for the thermal local problem in (10) was of $O(J_T\mathcal{T})$, here it grows to $O(\frac{2^{K_j}}{K_j}J_T\mathcal{T})$. Nevertheless, this is still not a significant increase, because $K_j \leq 6$ for the Brazilian case. We report on relative CPU times for each local problem at the end of Section 6 below. From these times it can be seen that the bulk of one evaluation of the augmented dual function Θ_c is concentrated on the local hydro-thermal problem.

Finally, to include start-up and shutdown times constraints we again duplicate variables and dualize the corresponding artificial constraint. Ramp constraints can be included likewise, along the lines of [14].

6. Numerical results

Numerical experiments are organized in three parts. First, we analyze the consistency of the method. Specifically, for a battery of ten test problems of increasing complexity, we compare the results obtained using our approach with those obtained with the previous continuous model. Comparisons are possible if thermal UC constraints are omitted (i.e., if there are only continuous variables, and no minimum generation or ramp constraints).

For the second part, we focus on one particular test problem and analyze different schedules obtained for configurations corresponding to the inclusion of various thermal UC constraints.

In the third part, we analyze the numerical behavior of the general methodology presented in this paper, with a special focus on the good numerical behavior of bundle methods. We start by compare computational times spent in the bundle and primal recovery phases, as well individual times for the corresponding local problems. We also analyze the performance of both bundle and the augmented Lagrangian method with respect to different precisions. Finally, to assess the quality of the maximization step, we compare our results with those obtained when using, instead of a bundle method, a subgradient method as in (12).

6.1. Consistency analysis

Table I on page 16 shows, for 9 different test configurations, the number of hydraulic and thermal plants, thermal units and percentages of hydraulic and thermal power capacity in MW.

Table I. Test problems.

Configuration	1	2	3	4	5	6	7	8	9
$\mathcal{N}_{\mathtt{H}}$	9	16	24	32	33	40	47	52	61
$\mathcal{N}_{\mathtt{T}}$	5	8	9	12	12	12	12	12	15
# thermal units	14	19	20	25	25	25	25	25	29
$\%\mathrm{MW}_{\mathtt{H}}$	89.1	88.5	93.1	92.1	94.5	95.3	95.9	96.1	96.1
$\%\mathrm{MW}_{\mathtt{T}}$	10.9	11.5	6.9	7.9	5.5	4.7	4.1	3.9	3.9

Table II on page 17 shows, for the battery of problems in Table I, the results obtained with the continuous model (`ContMod`), with the bundle method in § 4 (`BundleDual`), and with the primal recovery (`PrimRec`).

`ContMod` uses an optimality tolerance of 0.001% of the starting cost. Future value functions $\mathcal{C}_{\mathtt{H}}$ generated by `ContMod` are used in the dual problems (11) and (16).

Unless specified otherwise, the bundle phase of resolution is stopped with tolerances $\mathtt{tol}_\varepsilon = 10^{-7}$ and $\mathtt{tol}_G = 10^{-4}$, starting with λ^0 equal to 1.1 times the linear part of $\mathcal{C}_{\mathtt{T}}$ for the thermic units and 0 for hidraulic units. Also, the maximum number of elements in the bundle is MAXBUN= 100.

The primal recovery step maximizing (17) for increasing values of c is denoted by `PrimRec`. We stop the method when

$$\|p - q\| = \|(p_{\mathtt{T}}, p_{\mathtt{H}}) - (q_{\mathtt{T}}, q_{\mathtt{H}})\| \leq \mathtt{tol}_{\mathtt{PrimRec}} \tag{18}$$

with $\mathtt{tol}_{\mathtt{PrimRec}} = 0.1$. The initial values for $\lambda_{A_{\mathtt{T}}}$, p and q are the final optimal values $\hat{\lambda}_A$, $\hat{p}$ and $\hat{q}$ in `BundleDual`. The starting value of $\lambda_{A_{\mathtt{H}}}$ is given by the final marginal hydraulic cost obtained with `ContMod`.

Table II. Differences in final optimal costs (absolute values).

	factor	1	2	3	4	5	6	7	8	9
`ContModvs.BundleDual`	10^{-7}	0.983	3.31	25.7	24.16	0.792	33.4	26.1	65.2	37.9
`ContModvs.PrimRec`	10^{-7}	7.34	3.58	57.5	23.6	2.30	33.2	23.1	57.3	32.9
`BundleDualvs.PrimRec`	10^{-7}	8.32	0.27	83.2	0.504	1.51	0.22	3.05	7.91	5.03

Table II shows (in absolute values) the differences of optimal costs obtained with the three techniques, compared two by two. Since there are no UC constraints, the primal recovery phase should not alter the optimal solution found by `BundleDual`, and this is indeed the case, up to a minor difference less than 10^{-6} in all cases. Similar behaviour is observed in all the comparisons, even though optimal costs are high figures, ranging between 1.21 10^{10} and 1.52 10^{10}.

6.2. Operational analysis

We now focus on Configuration 1 in Table I, representing part of the Brazilian south region, where the majority of thermal plants is located. It has 9 hydro-plants with 7118 MW of total installed capacity and 5 thermal plants with 14 units with total capacity of 874 MW. With respect to percentages displayed in Table I, we mention that the effective thermal participation of this configuration in the dispatch is bigger than 10.9%. This is because some peculiarities on reservoirs make it impossible for the hydraulic mix to attain its total capacity. In addition, thermal units are not constrained to fuel shortage.

The demand load D^t is distributed in three increasing levels along the day

low	from 0h to 6h	1000MW
medium	from 6h to 18h and from 21h to 0h	2000MW
high	from 18h to 21h	3000MW

Table III of page 18 shows the various operational constraints considered for the thermal plants and units. Operating costs have the form $\mathcal{C}_{\mathtt{T}}(x_{\mathtt{T}}) = \mathcal{C}_{\mathtt{T}fix} + \mathcal{C}_{\mathtt{T}lin} x_{\mathtt{T}} + \mathcal{C}_{\mathtt{T}qua} {x_{\mathtt{T}}}^2$. Minimum and maximum generations are expressed in MW, costs in U\$S (per hour, MWh or MW^{2}h); and start-up and shutdown times in hours. Some of the data in Table III are fictitious, whether for lack of data or to force our model to deal with rigid thermal constraints. In particular, for units with same operational costs, we impose different UC constraints and analyze the effect in the result, when compared to the optimal generations found by `ContMod` (without UC constraints).

Table III. Different UC constraints.

Plant	9			11		13				14		16
unit	1	2	3	1	2	1	2	3	4	1	2	1
$\underline{x_T}$	40	40	40	30	30	0	0	0	0	0	0	0
$\overline{x_T}$	160	160	160	116	116	18	18	18	18	33	33	24
State$_0$	ON	ON	OFF	OFF	OFF	ON	ON	ON	ON	OFF	OFF	ON
$\mathcal{C}_{T\,fix}$	0	0	0	0	0	0	0	10	10	2.0	2.0	0
$\mathcal{C}_{T\,lin}$	22.32	22.32	22.32	48.92	48.92	53.9	53.9	53.9	53.9	62.52	62.52	70.00
$\mathcal{C}_{T\,qua}$	0	0	0	0	0	0	0	1	1	0	0	0
$\mathcal{C}_{T}up$	0	0	0	40000	0	0	0	0	0	0	0	0
t^{up}	6	6	6	4	4	1	1	1	1	2	2	3
t^{down}	6	6	6	4	4	1	1	1	1	2	2	3

Figure 6.2 in page 18 shows the operation of the 3 identical units composing plant # 9, the cheaper thermal plant in the configuration. We see that the two first units, initially on line, continued operating during the whole period at their maximum capacity. The third unit, initially off line, started its generation as soon as allowed by the minimum start up time constraint.

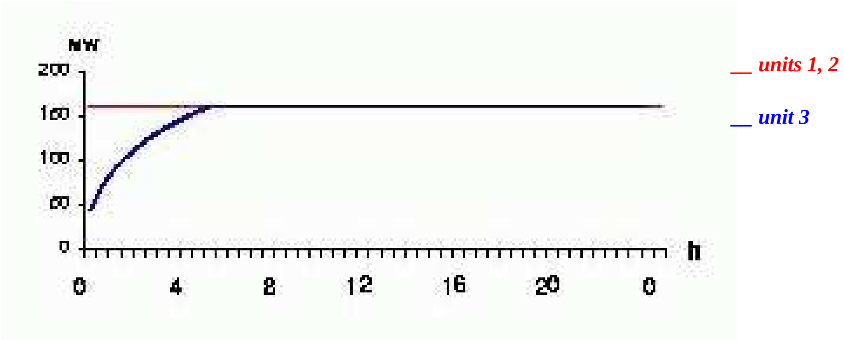

Figure 1. Generation schedule of plant # 9.

We do not display the operating schedule of plant #11. Note, however, that both units in this plant have identical characteristics, except for a high start-up cost assigned to unit 1. This high cost made the optimization process not to turn on unit 1 at all. The total production of plant #11 is the generation of unit 2, which presents a similar behaviour to the 3rd unit in plant #9.

Figure 6.2 in page 19 shows the operation of unit 3 in plant #13. Note that the introduction of a small quadratic cost eliminates the bang-bang phenomenon of oscillation between the minimum and maximum generation values. In addition, there is a clear increase of generated power during the period of high demand.

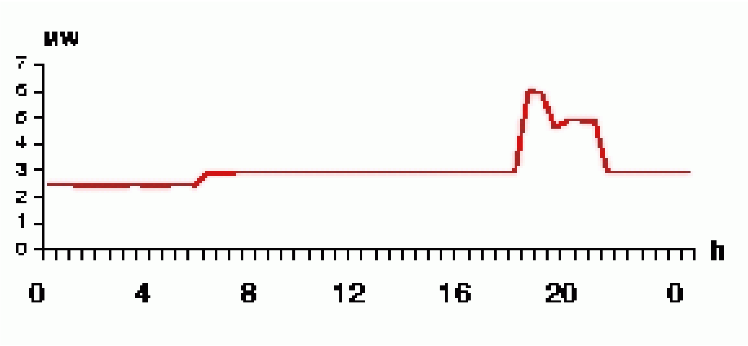

Figure 2. Generation schedule of unit 3, plant # 13.

Finally, Figure 3 in page 19 shows the operation of any of the units in plant #14, where we introduced fixed costs. In this case, the model decided to turn on the units only when there are peaks of high demand.

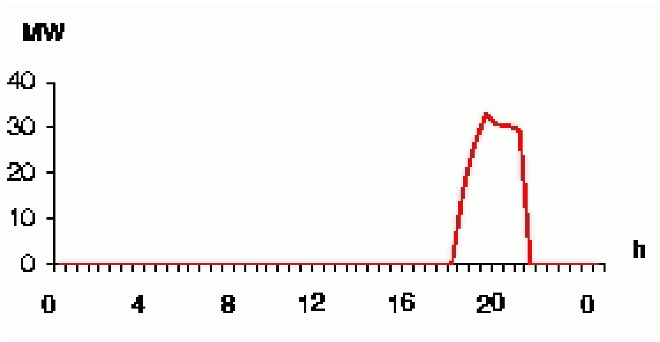

Figure 3. Generation schedule of plant # 14.

6.3. Methodological analysis

In this part, we first analyze the sensitivity of the general approach to different parameters, such as stopping tolerances of each phase and starting points.

We also give a qualitative comparison between subgradient and bundle methods in the maximization dual step.

6.3.1. *General approach*

Table IV in page 20 shows CPU times (in minutes and seconds) of the bundle iteration `BundleDual` in various situations. We check two starting points, namely, $\lambda^0 = 1.1$ the linear part of $\mathcal{C}_T$ and $\lambda^0 = 0.0$; different bundle sizes MAXBUN, and different stopping tolerances.

Table IV. CPU times (min:sec) for `BundleDual`.

	1	2	3	4	5	6	7	8	9
$\lambda^0 = 1.1\mathcal{C}_T$	0:22	2:00	6:58	1:08	1:13	2:28	1:10	1:19	2:30
$\lambda^0 = 0$	0:25	9:36	2:45	1:37	1:27	2:03	1:53	2:16	2:35
MAXBUN=3	0:09	6:44	24:03	1:45	2:18	1:31	3:18	2:48	8:21
MAXBUN=8	0:12	3:47	3:20	1:09	1:09	1:44	1:07	1:10	2:41
MAXBUN=100	0:22	2:00	6:58	1:08	1:13	2:28	1:10	1:19	2:30
$\mathtt{tol}_\varepsilon = 10^{-4}$ $\mathtt{tol}_G = 10^{-2}$	0:11	0:42	2:28	0:49	1:12	1:15	1:03	1:13	2:56
$\mathtt{tol}_\varepsilon = 10^{-7}$ $\mathtt{tol}_G = 10^{-4}$	0:22	2:00	6:58	1:08	1:13	2:28	1:10	1:19	2:30
$\mathtt{tol}_\varepsilon = 10^{-10}$ $\mathtt{tol}_G = 10^{-6}$	0:40	1:38	6:09	0:58	1:10	2:14	1:03	1:17	2:43

An important remark concerning the different cases in Table IV is that, for each configuration, the optimal dual costs are the same. This fact confirms the *robustness* of bundle methods. Namely, our bundle method always provides the same solution, because of the reliable stopping test. Of course, the CPU time needed to obtain such solutions is smaller with better starting points ($\lambda^0 = 1.1\mathcal{C}_T$) and with big enough bundles. In this respect, it is interesting to note that even though the fact of having a big bundle (MAXBUN=100) makes each iteration heavier (the size of (14) is bigger), the algorithm needs less iterations to get to a solution than when using smaller bundles (MAXBUN=3,8). As a result, total CPU times are usually smaller for bigger bundles. In addition, observe that complex configurations such as # 8, #9 in Table I are solved in less than 3 minutes. Finally, note that our initial choice of stopping tolerances ($\mathtt{tol}_\varepsilon = 10^{-7}, \mathtt{tol}_G = 10^{-4}$) seems adequate.

For Configuration 1 in Table I, we check the stopping criterion (18) for the Augmented Lagrangian phase `PrimRec`. Table V in page 21 shows optimal costs (in thousands of dollars) obtained for different values of the tolerance $\mathtt{tol}_{\mathtt{PrimRec}}$.

Table V. Checking stopping criterion for PrimRec.

$\mathtt{tol}_{\mathtt{PrimRec}}$	Operation cost	CPU time (min)	# iterations
10^{2}	15.256.356,782	2.13	25
10^{1}	15.256.355,427	2.52	29
10^{0}	15.256.321,201	3.45	38
10^{-1}	15.256.320,935	5.49	56
10^{-2}	15.256.320,926	8.60	79
10^{-3}	15.256.320,902	24.23	123

In terms of gain of precision, we conclude that, when compared to the extra number of iterations involved, it is not worthy to use tolerances smaller than 0.1.

For the same example (Configuration 1, $\mathtt{tol}_{\mathtt{PrimRec}} = 10^{-1}$) we compare relative times spent by each phase of the model. We denote by H and T the local problems (9) and (10), respectively. Local problems in the PrimRec phase are denoted by HT, T and H, according to their order in (17). Finally the time spent by the bundle iteration (14) is denoted by B:

% time BundleDual			% time PrimRec		
% H	% T	% B	% HT	% T	% H
65.51	0.19	7.90	17.61	0.03	8.75
total %	73.61		total %	26.39	

We see that in both phases most of the time is spent solving the local problems with constraint set $\mathcal{S}_{\mathtt{D1}}$. We are currently considering how to further decompose the demand set. Keeping in mind that the future cost function $\mathcal{C}_{\mathtt{H}}$ is not decomposable, this is a subject that requires some investigation.

Since, typically, the bundle phase finishes very soon, we prefer to spend more time in the bundle phase BundleDual. We observe that entering the phase PrimRec with a closer starting point reduces total computing times. Our assertion is also based on the fact that local problems in BundleDual are easier to solve than those in PrimRec.

6.3.2. *Bundle versus subgradient methods*

In Table VI in page 22 we compare the effect of making the dual maximization step with bundle and subgradient methods.

For the subgradient method, we use the update (12), with stopping test given by

$$\|g(\lambda_n)\| \leq \mathtt{tol}_G \quad \text{for } \mathtt{tol}_G = 10^{-4}.$$

We also use the emergency stopping test of maximum number of iterations, set to 500.

Numerical results in Table VI confirm what announced by theory: the apparently simpler subgradient method struggles to achieve high accuracy. Even when the level of accuracy is comparable (for instance for configurations # 2, # 7, # 8, # 9), the subgradient method spends much more CPU time than our bundle method.

Table VI. `BundleDual`vs `SubGrad`. Optimal dual values and CPU times

Configuration	`BundleDual` dual cost	`SubGrad` dual cost	`BundleDual` min:sec	`SubGrad` min:sec
1	15.256.449.399	15.256.449.399	0:40	1:47
2	14.822.203.286	14.822.203.286	2:00	23:15
3	14.166.320.355	14.166.320.858	6:58	21:03
4	13.885.710.295	13.885.710.295	1:08	5:58
5	13.882.948.367	13.882.948.367	1:13	2:33
6	13.626.288.294	13.626.288.294	2:28	9:48
7	12.797.701.970	12.797.701.970	1:10	4:06
8	12.637.734.322	12.637.734.322	1:19	4:23
9	12.129.155.717	12.129.155.717	2:30	6:06

Final Remarks

We presented a Lagrangian relaxation approach to include the thermal unit commitment in the Brazilian power system. The dual maximization step is done using a bundle method, which has some outstanding features, particularly important for our large-scale complex problem. Our numerical experience, based on non trivial examples of the Brazilian power system confirms what predicted by the theory.

In order to allow a quick resolution of (14), whose dual has dimension MAXBUN at most, we use bundles of size MAXBUN$\leq$ 100 and warm starts. Hence, the bundle phase typically finishes in less than 3 minutes.

In addition, our bundle method is robust and reliable with respect to different starting points, stopping tolerances and bundle sizes. We also observe that, in terms of total CPU times, it is preferable to have bigger bundles.

When compared to cutting-planes methods, our algorithm allows to aggregate the information in the bundle to avoid the typical memory overflows of cutting-planes, due to an infinite accumulation of the affine functions defining the model $\hat{\Theta}$. This point is crucial when the dimension of λ is large, like in our application.

Finally, our numerical results show that when high accuracy is a must, subgradient methods spend much more CPU time than bundle methods.

In summary, the combination of bundle methods and inexact augmented Lagrangians seems a very suitable tool for obtaining reliable and accurate marginal prices with low CPU times.

References

1. Bacaud, L., C. Lemaréchal, A. Renaud, and C. Sagastizábal: 2001, 'Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners'. *Computational Optimization and Applications* **20**(3), 227–244.
2. Batut, J. and A. Renaud: 1992, 'Daily Generation Scheduling with Transmission Constraints: a New Class of Algorithms'. *IEEE Transactions on Power Systems* **7**, 982–989.
3. Bertsekas, D.: 1995, *Nonlinear Programming*. Athena Scientific.
4. Blondel, H. and A. Renaud: 1991, 'Décomposition par les prix et Lagrangien augmenté: une association efficace pour le traitement des problèmes de production'. *EDF Bull. Direction Études Rech. Sér. C Math. Inform.* **4**, iii, 23–43 (1992).
5. Bonnans, J., J. Gilbert, C. Lemaréchal, and C. Sagastizábal: 2002, *Numerical Optimization. Theoretical and Practical Aspects*, Universitext. Berlin: Springer-Verlag. to appear.
6. Casas, E. and C. Pola: 1993, 'An algorithm for indefinite quadratic programming based on a partial Cholesky factorization'. *RAIRO Rech. Opér.* **27**(4), 401–426.
7. Cheney, E. and A. Goldstein: 1959, 'Newton's method for convex programming and Tchebycheff approximations'. *Numerische Mathematik* **1**, 253–268.
8. Cohen, G. and D. L. Zhu: 1984, 'Decomposition coordination methods in large scale optimization problems: the nondifferentiable case and the use of augmented Lagrangians'. In: *Advances in large scale systems, Vol. 1*. Greenwich, CT: JAI, pp. 203–266.
9. Conejo, A. J. and N. J. Redondo: 1999, 'Short-term hydro-thermal coordination by Lagrangian relaxation: solution of the dual problem'. *IEEE Transactions on Power Systems* **14**(1), 89–95.
10. Culioli, J.-C. and G. Cohen: 1990, 'Decomposition/coordination algorithms in stochastic optimization'. *SIAM J. Control Optim.* **28**(6), 1372–1403.
11. Feltenmark, S. and K. C. Kiwiel: 2000, 'Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems'. *SIAM J. Optim.* **10**(3), 697–721 (electronic).
12. Goffin, J.: 1977, 'On convergence rates of subgradient optimization methods'. *Math. Program.* **13**, 329–347.
13. Kelley, J. E.: 1960, 'The cutting plane method for solving convex programs'. *J. Soc. Indust. Appl. Math.* **8**, 703–712.
14. Lai, S.-Y. and R. Baldick: 1999, 'Unit Commitment with Ramp Multipliers'. *IEEE Transactions on Power Systems* **14**(1), 58–64.
15. Lemaréchal, C., F. Pellegrino, A. Renaud, and C. Sagastizábal: 1996, 'Bundle methods applied to the unit-commitment problem'. In: J. Doležal and J. Fidler (eds.): *System Modelling and Optimization*. pp. 395–402.
16. Lemaréchal, C. and A. Renaud: 2001, 'A geometric study of duality gaps, with applications'. *Math. Program.* **90**(3, Ser. A), 399–427.
17. Lemaréchal, C. and C. Sagastizábal: 1997, 'Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries'. *SIAM Journal on Optimization* **7**(2), 367–385.
18. Maceira, M., C. Mercio, B. Gorenstin, S. Cunha, C. Suanno, M. Sacramento, and A. Kligerman: 1998, 'Aplicacão do modelo NEWAVE na avaliacão energética das interligacões dos sistemas norte/nordeste e sul/sudeste brasileiro'. In: *VI Symposium of specialists in eletric operational and expansion planning*. Electronic version (CD).
19. Maceira, M., L. Terry, A. Diniz, L. de Souza, F. Costa, S. Romero, S. Binato, S. Amado, C. Vilasboas, and R. Vilanova: 2000, 'Despacho de Geracão horário com representacão detalhada de restricões hidráulicas'. In: *VII Symposium of specialists in eletric operational and expansion planning*. SP-063, Electronic version (CD).
20. Merlin, A. and P. Sandrin: 1983, 'A New Method for Unit Commitment at Électricité de France'. *IEEE Transactions on Power Aparatus and Systems* **PAS-102**(5), 1218–1225.
21. M.Guignard and S. Kim: 1987, 'Lagrangean decomposition: a model yielding stronger lagrangean bounds'. *Math. Program.* **39**(2), 215–228.
22. Pereira, M. V. F. and L. M. V. G. Pinto: 1991, 'Multi-stage stochastic optimization applied to energy planning'. *Mathematical Programming* **52** pp. 359–375.

23. Sheble, G. B. and G. Fahd: 1994, 'Unit commitment literature synopsis'. *IEEE Transactions on Power Systems* **9**(1), 128–135.
24. Shor, N.: 1985, *Minimization methods for non-differentiable functions*. Springer-Verlag, Berlin.
25. Zhuang, G. and F. Galiana: 1988, 'Towards a more rigourous and practical unit commitment by Lagrangian relaxation'. *IEEE Trans. Power Systems* **3**, 763–773.