Math. Program., Ser. A 104, 129–151 (2005)

Digital Object Identifier (DOI) 10.1007/s10107-005-0593-4

L. Dubost · R. Gonzalez · C. Lemaréchal

A primal-proximal heuristic applied to the French Unit-commitment problem

Received: October 16, 2003 / Accepted: February 15, 2005
Published online: April 28, 2005 –

Abstract. This paper is devoted to the numerical resolution of unit-commitment problems, with emphasis on the French model optimizing the daily production of electricity. The solution process has two phases. First a Lagrangian relaxation solves the dual to find a lower bound; it also gives a primal relaxed solution. We then propose to use the latter in the second phase, for a heuristic resolution based on a *primal proximal* algorithm. This second step comes as an alternative to an earlier approach, based on augmented Lagrangian (i.e. a *dual* proximal algorithm). We illustrate the method with some real-life numerical results. A companion paper is devoted to a theoretical study of the heuristic in the second phase.

Key words. Unit-commitment problem – Proximal algorithm – Lagrangian relaxation – Primal-dual heuristics – Combinatorial optimization

1. Introduction

This paper reports on a practical application: to optimize the daily production of the French electricity mix. We symbolize this *unit-commitment* problem by

$$\min c(x)\,, \quad x \in X \subset \mathbb{R}^n\,, \quad g(x) = 0 \in \mathbb{R}^m\,, \tag{1.1}$$

and Section 2 outlines more precisely the model currently in operation at EdF (Électricité de France, the French Electricity Board).

In (1.1), $X = \prod_i X^i$, $c(x) = \sum_i c^i(x^i)$, where i denotes a production unit (nuclear, thermal or hydro-valley), with decision variable x^i varying in X^i; also, $g(x) = \sum_i g^i(x^i)$ collects *linking constraints*: the problem is decomposable. It is therefore conveniently solved via Lagrangian relaxation (see [8, 4, 15, 35] among others, and also the review [42]): to (1.1), is associated the Lagrangian problem

$$\theta(\lambda) := \min_{x\in X} L(x,\lambda)\,, \qquad \text{where}$$
$$L(x,\lambda) := c(x) + \lambda^\top g(x)\,. \tag{1.2}$$

L. Dubost: EDF R&D, 1 avenue du Général de Gaulle, 92141 Clamart, France.
e-mail: louis.dubost@edf.fr

R. Gonzalez: EDF, 9 route de la porte de Buc, 78005 Versailles, France.
e-mail: robert.gonzalez@rte-france.com

C. Lemaréchal: INRIA, 655 avenue de l'Europe, Montbonnot, 38334 Saint Ismier, France.
e-mail: claude.lemarechal@inrialpes.fr

Mathematics Subject Classification (2000): 90B30, 90C06, 90C90

Computing $\theta(\lambda)$ amounts to solving independently the local problems

$$\min_{x^i \in X^i} c^i(x^i) + \lambda^\top g^i(x^i)\,, \tag{1.3}$$

where i ranges over the set of production units.

A dual algorithm constructs iteratively λ_k to maximize θ (the dual problem), (1.3) being repeatedly solved for $\lambda = \lambda_k$. Section 3 recalls the approach and gives some illustrative results obtained with EdF's unit-commitment model, in which a bundle method maximizes θ [32], with the help of the quadratic solver of [27, 28].

The above technique provides a lower bound for the optimal cost. However, (1.1) itself is still unsolved: no optimal schedule need be produced while maximizing θ, not even an approximately feasible one. Several works are devoted to the resulting *primal recovery* problem via appropriate heuristics; see [42]. In its current implementation, EdF uses a heuristic based on augmented Lagrangian. The heuristics presented in the present paper is closer to [16], and can roughly be outlined as follows.

After convergence, the dual algorithm produces a primal point $\hat{x}$, which solves a certain *convex relaxation* of (1.1) (think of column generation, see [14, 36], [30, Sections 2.6, 4.2, 5.2], [47]). We call this $\hat{x}$ *pseudo-schedule*; unless there is no duality gap, the pseudo-schedule is a non-trivial convex combination of schedules, and therefore does not lie in the nonconvex set X. Nevertheless, $\hat{x}$ tends to satisfy the linking constraints, a property which is enjoyed by none of the minimizers in (1.3) – here precisely lies the trouble with Lagrangian relaxation!

Starting from this idea, we perturb θ to

$$\theta_r(\lambda) := \min_{x \in X} L_r(x, \lambda)\,, \qquad \text{where}$$
$$L_r(x, \lambda) := L(x, \lambda) + r\|x - \hat{x}\|^2\,.$$

A crucial observation is that θ_r is readily computable because the decomposability (1.2)–(1.3) is preserved:

$$L_r(x, \lambda) = \sum_i \big[c_i(x^i) + \lambda^\top g^i(x^i) + r\|x^i - \hat{x}^i\|^2\big]\,.$$

It can therefore be maximized with just the same bundle algorithm that maximized θ, using essentially the same subalgorithms that solved (1.3). Section 4 presents the heuristics and gives illustrative results suggesting the superiority of the present approach over augmented Lagrangian.

Actually, the above heuristics gives fairly good empirical results. Besides, it is not limited to unit-commitment: it can be applied to most nonconvex problems lending themselves to Lagrangian relaxation (the only restriction being that the local solvers (1.3) must accommodate simple quadratic costs). A question of interest is therefore: can it be assessed by some theoretical study? This is the subject of [13], a sequel to the present work; it suggests that, unfortunately, little can be said along these lines.

2. The French model for electricity production

The French production mix is composed of nearly 150 thermal units and 50 hydro-valleys. Depending on the period of the day and on the season of the year, the load demand

oscillates between some 40 000 and 70 000 MW. A reasonably reliable demand forecast, as well as the list of available power units with their updated technical constraints and economic characteristics, are available each day by 4 pm. Appropriate production schedules must then be computed, to be sent to the local units by 5 pm.

The generation mix is composed of nuclear, thermal (fuel or coal) and hydro-plants, which represent respectively 61%, 16% and 23% in terms of total capacity. Of course these figures are very different from the actual distribution of the generation, which highly depends on the economic characteristics of each unit. Gas turbines are also present; they are not included in the model: due to their high operating costs and flexibility, they are generally viewed as a potential reserve, to be used in some emergency cases.

Calling T the time horizon, each schedule is represented by a vector $x^i \in \mathbb{R}^T$; each day is discretized in 48 half-hours. However, optimizing over one day only (which is the required exercise) is a myopic strategy. It may produce costly situations such as shutting down by the end of the day some thermal unit which will have to be switched on early the day after: a startup sequence is a costly transition. This is why the model takes $T = 96$, i.e. the optimization is performed over two days, even though only the first half of the schedules are actually sent to the local units.

Altogether, the unit-commitment problem is to compute feasible schedules x^i, minimizing the total operating cost over the period, while matching the predicted demand and satisfying some *reserve* constraints, present to face unexpected events (increase in demand, failure of a production unit). We now proceed to describe the technological and economic characteristics of EdF's model. See also [11] for other possible "real-life" models.

2.1. Nuclear and thermal units

Nuclear and thermal units have similar behaviours; their decision variable x^i used in the notation (1.1) is indeed the power delivered p^i.

The operating domain related to each thermal unit is defined by a particular set of dynamic constraints, a non-exhaustive list of which is given below:

(i) The delay between two output variations must be longer than some value depending on the unit.
(ii) Ramping constraints are associated to each thermal unit: jumping from one output level to another cannot be done instantaneously.
(iii) After switching off a unit, a mimimum down time must precede any following startup sequence.
(iv) Particular startup curves must be used, depending on which cold start they apply to (e.g. depending on how long the unit has been switched off).
(v) When a generator is on, it must remain online for a minimum amount of time.
(vi) When a generator is on, its output level must remain between given maximum and minimum values (both positive); say $0 < \underline{p}^i \leqslant p^i \leqslant \bar{p}^i$.
(vii) A generator may or may not be allowed to participate in the frequency regulation at time t (this is a device to ensure the automatically generated reserves, see §2.3 below).

If $p^i = \left(p_1^i, \ldots, p_T^i\right)$ denotes the production vector of unit i, satisfaction of these various constraints can be symbolized by $p^i \in P^i$. Besides, a cost $c^i(p^i)$ is associated to the production vector (with respect to the notation (1.1), (1.3), remember that the decision variable is here $x^i = p^i$). When the linking constraints of §2.3 are dualized, unit i will aim at minimizing $c^i(p^i) + \lambda^\top g^i(p^i)$, subject to $p^i \in P^i$: a fairly complicated problem, with lots of Boolean constraints, and nonlinear cost.

A simplifying assumption is made at this point, replacing the continuous segment $[\underline{p}^i, \bar{p}^i]$ introduced in (vi) by a set of discrete values: for instance, the output level of a thermal unit can take on 3 values only ($\underline{p}^i$, $\bar{p}^i$, and a nominal value in between). With this simplification, the problem becomes purely discrete, and can be given a formulation suitable to standard *dynamic programming*. For this, a number of *state variables* are used, to characterize the state of unit i; for example

(i) the time q_t^i elapsed since the last output variation,
(iii) when the unit is off, the time r_t^i elapsed since the switch off (this is also useful to cope with (iv)),
(v) when the unit is on, the time s_t^i elapsed since the switch on,
(. . .) and so on.

These state variables evolve along time according to appropriate *state equations*, in which p^i acts as a *control variable*. The states are themselves subject to various constraints, such as $q_t^i \geqslant \underline{q}^i$, $r_t^i \geqslant \underline{r}^i$, $s_t^i \geqslant \underline{s}^i$, and so on. In a word, the problem can be solved efficiently by dynamic programming.

Remark 21. As already said, our list of constraints above is not exhaustive, the main merit of dynamic programming being its flexibility. However there are also *daily constraints*, expressing for example that the number of output variations and startup sequences in a day is limited. A direct treatment of these constraints increases the number of states drastically, the cost in CPU becomes prohibitive. The current strategy is therefore to penalize or dualize them.

For the record, "semi-local" constraints also exist. The units are actually clustered in *production plants* (a plant may contain up to 4 units) and constraints link units within the same plant. For example, there is a minimum delay between two startup sequences of two different units within the same plant. No rigorous method has been found to deal with this kind of constraints; dualizing them would increase the size of the problem with little hope to compute schedules satisfying them, therefore they are ignored in the model. □

2.2. *Hydro-valleys*

The French hydro-power system is composed of 50 independent hydro-valleys. A hydro-valley i is a set of interconnected reservoirs – we index them by r – and associated power plants – indexed by j; the plants in turn have a certain number of turbines. The decision variable x^i in the notation (1.1) is now the set of flows w^j passing through the various plants, each of which results in a power $p^j = p^j(w^j)$. A flow w^j can take on finitely many values; these values correspond to the optimal rate of efficiency of the turbines in

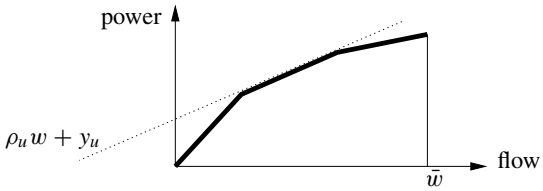

Fig. 2.1. Hydro-production as a function of flow

the given plant. Besides, some plants are equipped to pump water from their downstream reservoir up into their upstream reservoir (thus allowing to re-use water later in peak hours).

Again a simplifying assumption is made here: each w^j is allowed to vary continuously in a given interval[1] $[0, \bar{w}^j]$. Besides, the rate of each turbine is considered as fixed. With this simplification, the power delivered is a piecewise linear function, which turns out to be *concave*, as depicted on Fig. 2.1 (each break point corresponds to switching a turbine, and the most efficient turbines are of course switched on first). As a result, the power delivered can be written as the minimum of a number of affine functions; say

$$p^j(w^j) = \min\{\rho_g^j w^j + y_g^j : g \text{ runs over the turbines}\}. \tag{2.1}$$

Pumping is modelled as follows: every plant with the pumping capability is duplicated in two plants, say j and j'; the (nonnegative) flow $w^{j'}$ of the "dummy" plant j' is counted upstream, and $p^{j'}(w^{j'}) \leqslant 0$ (pumping consumes energy!).

The volume V^r in a given reservoir r is then a *state variable*, which obeys an equation of the type

$$V_t^r = V_{t-1}^r - \sum_{j \in N_\downarrow(r)} w_{t+\tau(r,j)}^j + \sum_{j \in N_\uparrow(r)} w_{t-\tau(j,r)}^j + O_t^r, \quad t = 1, \ldots, T.$$

Here O represents the exchange of water with the outside (rain, snow, spillage); $N_\downarrow(r)$ [resp. $N_\uparrow(r)$] is the set of neighboring plants down [resp. up] reservoir r; $\tau(r, j)$ [resp. $\tau(j, r)$] is the travel time of water from reservoir r to plant j [resp. from plant j to reservoir r] – with the appropriate conventions for the dummy pumps.

Now the cost of the schedules in valley i is just the global loss in water:

$$c^i(w^i) = \sum_r \gamma^r (V_0^r - V_T^r), \quad \text{for each valley } i;$$

this is a linear function of the decision variable w. The marginal costs γ^r result from mid- and long-term models, assessing the value of water. To complete the model, bound constraints are introduced on the volumes V^r.

Some crucial observations are relevant at this point.

[1] Compare this simplification with that of thermal plants, where continuous variables were replaced by discrete ones. After the simplified model is solved, a smoothing "off-model" phase is performed, aimed at restoring feasible w^j's without changing the overall power produced.

- The rate ρ_u^j of a given turbine does not depend on the volume in the reservoir feeding it; in other words: water elevation in the reservoirs is considered as a constant over the period $[0, T]$, thus allowing the simple expression (2.1). Due to the shortness of the optimized period (max. 2 days) and to the existence of bound constraints on the reservoir's volumes, this is quite a valid approximation.
- A plant cannot discharge and pump simultaneously: there are constraints of the form $w_t^j w_t^{j'} = 0$. However, it can be shown that these constraints are automatically satisfied by an optimal solution – such a property relies heavily on the particular form of the Lagrangian, though.
- As will be seen in §2.3, dualization of the constraints results in a hydraulic Lagrangian of the form
$$c^i(w^i) - \lambda^\top \sum_j p^j(w^j)\,.$$
If $\lambda \geqslant 0$ – quite a natural property, always true in practice –, the particular form (2.1) of $w \mapsto p(w)$ allows an LP formulation, via the addition of an extra variable for each unit u.

As a result "everything is linear": minimizing the Lagrangian over the feasible set amounts to minimizing a linear function over a polyhedron, which we symbolize by $w^i \in W^i$. The resulting linear program is solved by an interior-point method, as described in [45] and implemented in [19]. In view of the form of the constraints, an LP solver tailored to flow problems would be more appropriate. However, some valleys might have additional constraints (for example limitations on flow variations); besides, quadratic Lagrangians will appear in §4. This motivates the choice of an interior-point methodology.

In what follows, we will use the notation ϑ [resp. $\mathcal{H}$] for the thermal [rep. hydraulic] mix. The total cost of a schedule is then

$$\sum_{i\in\vartheta} c^i(p^i) + \sum_{i\in\mathcal{H}} c^i(w^i)\,. \tag{2.2}$$

2.3. Linking constraints

Demand. Denote by $d^0 \in \mathbb{R}^T$ the predicted power demand vector. The mismatch between production and demand is

$$p_t^0 := d_t^0 - \sum_{i\in\vartheta\cup\mathcal{H}} p_t^i\,, \quad \text{for } t = 1, \ldots, T\,. \tag{2.3}$$

This notation introduces a "fictitious power generator" $p^0 \in \mathbb{R}^T$, analogous to a slack variable. Imposing brutally the constraint $p^0 = 0$ – or even $p^0 \geqslant 0$ – would ignore various uncertainties present in the problem (the power demand vector d^0 is not 100% reliable, some production failures may occur, etc.). Rather, the model attaches a cost

$$c^0(p^0) = \sum_t c_t^0(p_t^0)$$

to the fictitious generation, where the elementary cost c_t^0 is depicted on Fig. 2.2. Then the balance constraint is just (2.3).

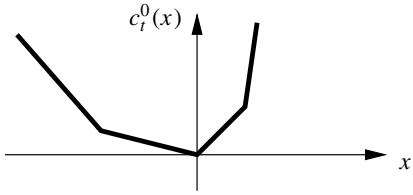

Fig. 2.2. Cost of the fictitious power generator

Reserves. In order to face unexpected events such as generator outages or sudden load variations in operational conditions, three types of *spinning reserves* impose constraints linking the production plants.

- The *primary* reserve continually adjusts (within a few seconds) small oscillations of the power load, simply by controlling automatically the angular speed of the alternators. This kind of reserve also aims at immediately compensating unexpected generator outages until the frequency of the network comes back to its nominal value. The total power thus available for adjustment must be at least some value $d^1 = (d_1^1, \dots, d_T^1)$.
- The *secondary* reserve also controls automatically the network's frequency with a response time of minutes, and must be at least some value d^2. The contribution of a given thermal unit i to the secondary reserve has a predefined value which depends on whether p^i is at its minimal, maximal or intermediate value.
- The *tertiary* reserve is triggered manually and its time response is longer than 20 minutes. The contribution of a thermal unit i to this reserve is $\bar{p}^i - p^i$, but there are a number of conditions; for example, it is 0 if the unit is following a decreasing ramp, or during a startup sequence, and of course if it is off.

The total reserve mismatch is then

$$p^{-k} := d^k - \sum_{i\in\vartheta} R^{k,i}(p^i) - \sum_{i\in\mathcal{H}} R^{k,i}\, p^i \in \mathbb{R}^T\,, \quad k = 1,2,3\,. \tag{2.4}$$

Each $R^{k,i}$ is a more or less complex function of the power; if i indexes hydraulic production, $R^{k,i}\, p^i$ is actually linear. Analogously to the power constraint (2.3), the notation above introduces three "fictitious reserve generators" $p^{-k} \in \mathbb{R}^T$, which induce three costs, as displayed on Fig. 2.3. Note that $c^{-k}(x) = 0$ for $x \geqslant 0$: in fact, $p^{-k} \geqslant 0$ is imposed as a hard constraint – even though it may be violated by the final solution, see Remark 22 below.

In summary, our commitment problem is

$$\begin{cases} \min \sum_{k=0}^{3} c^{-k}(p^{-k}) + \sum_{i\in\vartheta} c^i(p^i) + \sum_{i\in\mathcal{H}} c^i(w^i)\,; \\ p^i \in P^i\,,\ i \in \vartheta\,; \quad w^i \in W^i\,,\ i \in \mathcal{H}\,; \quad p^{-k} \geqslant 0 \in \mathbb{R}^T\,,\ k = 1,2,3\,; \\ d^0 - p^0 - \sum_{i\in\vartheta} p^i - \sum_{i\in\mathcal{H}} p^i(w^i) = 0 \in \mathbb{R}^T\,; \\ d^k - p^{-k} - \sum_{i\in\vartheta} R^{k,i}(p^i) - \sum_{i\in\mathcal{H}} R^{k,i}\, p^i(w^i) = 0 \in \mathbb{R}^T\,, \quad k = 1,2,3\,. \end{cases} \tag{2.5}$$

This problem is large-scale, heterogenous, mixed-integer, nonlinear; in a word: a nasty problem. However, dualizing the linking constraints is particularly attractive.

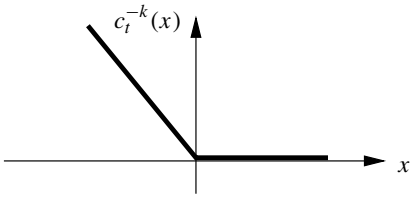

Fig. 2.3. Cost of a fictitious reserve generator

Remark 22 (Role of the fictitious costs). Solving this problem is clearly impossible. More precisely, it is impossible to obtain schedules satisfying exactly the various linking constraints (or rather the associated complementary slackness). A common objective function is therefore required to assess a given schedule, in terms of its real cost and of its constraint mismatch. The fictitious costs c^{-k} do the job: the value of a "real" schedule $\{p^i\}_{i\in\vartheta\cup\mathcal{H}}$ is the sum of its real cost (2.2) and of the total fictitious cost

$$c^0\Big(d^0 - \sum_{i\in\vartheta\cup\mathcal{H}} p^i\Big) + \sum_{k=1}^{3} c^{-k}\Big(d^k - \sum_{i\in\vartheta\cup\mathcal{H}} R^{k,i}(p^i)\Big)_- .$$

In particular, the fictitious cost of a reserve is useful to penalize violations of the constraints $p^{-k} \geqslant 0$, $k = 1, 2, 3$. This remark will be important for Sections 4 and 5. □

3. The dual problem and its resolution

A first step for the resolution of (2.5) via Lagrangian relaxation is to solve the dual. This provides a lower bound on the optimal cost. More importantly, a solution of a convexified form of (2.5) is also provided; we call it a pseudo-schedule, and it will play a crucial role to recover a primal heuristic solution in §4. The reader may wish to consult [30, 31] for a more complete account of duality, following the same lines.

3.1. The dual problem

Taking dual variables $\lambda \in \mathbb{R}^T$ (for the balance constraint (2.3) at each time period) and $\mu \in \mathbb{R}_+^{3T}$ (for each reserve constraint (2.4) at each time period), we form the lagrangian

$$\begin{aligned} L(p, w, \lambda, \mu) := {} & c^0(p^0) - \lambda^\top p^0 + \textstyle\sum_{k=1}^{3}\big(c^{-k}(p^{-k}) - \mu_k^\top p^{-k}\big) \\ & + \textstyle\sum_{i\in\vartheta} c^i(p^i) - \lambda^\top p^i - \mu^\top R^i(p^i) \\ & + \textstyle\sum_{i\in\mathcal{H}} c^i(w^i) - \lambda^\top p^i(w^i) - \mu^\top R^i p^i(w^i) \\ & + \lambda^\top d^0 + \mu^\top d\,. \end{aligned}$$

Its minimization for given λ, μ amounts to solving the local problems

$$\begin{aligned} &\min_{p^0\in\mathbb{R}^T} c^0(p^0) - \lambda^\top p^0\,, \\ &\min_{p^{-k}\in\mathbb{R}_+^T} c^{-k}(p^{-k}) - \mu_k^\top p^{-k} && k = 1, 2, 3\,, \\ &\min_{p^i\in P^i} c^i(p^i) - \lambda^\top p^i - \mu^\top R^i(p^i) && i \in \vartheta\,, \\ &\min_{w^i\in W^i} c^i(w^i) - \lambda^\top p^i(w^i) - \mu^\top R^i p^i(w^i) && i \in \mathcal{H}\,. \end{aligned} \tag{3.1}$$

The resulting optimal value is the dual function, which must be maximized over $(\lambda, \mu) \in \mathbb{R}^T \times \mathbb{R}^{3T}_+$.

From now on, we will return to the condensed notation (1.1), (1.2) and neglect the sign-constraint on the dual variables; then we have $x = (p^{-k}, p^{\vartheta}, w^{\mathcal{H}})$ and $m = 4T$.

It is well known that any dual value $\theta(\lambda)$ is a lower bound for the optimal cost in (1.1) = (2.5), and that θ is a concave function. Calling $x(\lambda)$ an optimal solution in (1.3) = (3.1), it is also known that the vector $g := g(x(\lambda)) \in \mathbb{R}^m$ is a subgradient of θ (or rather a "supergradient", θ being concave). Examining the local problems (3.1) shows that θ is a polyhedral function, with kinks at those λ such that $x(\lambda)$ is ambiguously defined. The dual problem is definitely a nonsmooth optimization problem.

The present dual approach is fairly classical, especially for unit commitment, and does not deserve much more comment. We just mention that the numerical illustrations to be reported below have been obtained with a bundle method. This is substantially more sophisticated than a mere subgradient method of the type $\lambda_+ = \lambda + sg(x(\lambda))$. The resulting dual algorithm therefore involves two (complicated but) very different worlds, sketched in Fig. 3.1: the primal world is in charge of solving (1.3) = (3.1), and the dual world iterates over the dual variable λ_k. For each of these two worlds, the other is a black box which delivers appropriate information upon request.

3.2. Overview of bundle methods

A bundle method works in a way analogous to column generation.

Having solved the local problems (1.3) for a number of dual iterates λ_k, one has obtained corresponding optimal solutions $x_k = x(\lambda_k)$; by construction, each x_k lies in the set X of "technically feasible" schedules; note that none of them satisfies the linking constraints, though – otherwise (1.1) would be solved. Then one constructs the "restricted" dual function

$$\hat{\theta}(\lambda) := \min_k L(x_k, \lambda) = \min_k \left[c(x_k) + \lambda^\top g(x_k)\right], \tag{3.2}$$

which (over-)approximates the actual θ, insofar as the "restricted" set $\{x_k\}_k$ approximates the whole of X.

Standard column generation finds the next iterate λ_+ by solving the LP restricted master $\max_\lambda \hat{\theta}(\lambda)$, i.e. $\max_\lambda \min_k L(x_k, \lambda)$, i.e.

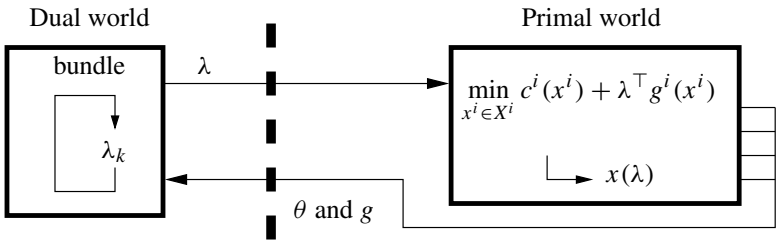

Fig. 3.1. The primal-dual dialogue

$$\begin{cases} \max\ r \quad (\lambda, r) \in \mathbb{R}^m \times \mathbb{R}\,, \\ r \leqslant c(x_k) + \lambda^\top g(x_k) \ \text{ for each } k\,. \end{cases} \tag{3.3}$$

In nonsmooth optimization, this is known as the method of Kelley, or Cheney-Goldstein, or cutting planes [10, 26].

Bundle methods are stabilized versions of column generation. At the current iteration, one chooses a stability center $\hat{\lambda}$, a stepsize $s > 0$, and one solves the QP restricted master $\max \hat{\theta}(\lambda) - \|\lambda - \hat{\lambda}\|^2/2s$, i.e.

$$\begin{cases} \max\ r - \frac{1}{2s}\|\lambda - \hat{\lambda}\|^2 \quad (\lambda, r) \in \mathbb{R}^m \times \mathbb{R}\,, \\ r \leqslant c(x_k) + \lambda^\top g(x_k) \ \text{ for each } k\,. \end{cases} \tag{3.4}$$

Without entering details, the stepsize is adjusted at each execution of the local problems; and the stability center $\hat{\lambda}$ is essentially the best among the previous iterates. It is this $\hat{\lambda}$ which is supposed to converge to some dual solution, say $\hat{\lambda} \to \lambda^*$.

3.3. Primal interpretation and pseudo-schedules

The above algorithms have a primal interpretation, obtained by (bi-)dualization. This is well-known for pure cutting planes: the dual of (3.3) is

$$\min_{\alpha \in \Delta} \sum_k \alpha_k c(x_k)\,, \quad \text{subject to} \quad \sum_k \alpha_k g(x_k) = 0\,, \tag{3.5}$$

where Δ is the unit simplex (of appropriate dimension, depending on the iteration number). In a bundle method, the QP restricted master (3.4) has also a dual, which turns out to be

$$\min_{\alpha \in \Delta} \sum_k \alpha_k c(x_k) + \frac{s}{2}\Big\|\sum_k \alpha_k g(x_k)\Big\|^2 + \hat{\lambda}^\top \sum_k \alpha_k g(x_k)\,. \tag{3.6}$$

Call $\hat{\alpha} \in \Delta$ an optimal solution of (3.5) or (3.6). This (bi)-dualization reveals important primal objects:

$$\begin{array}{ll} \text{a } \textit{pseudo-schedule} & \hat{x} := \sum_k \hat{\alpha}_k x_k \\ \text{with corresponding } \textit{pseudo-cost} & \hat{c} := \sum_k \hat{\alpha}_k c(x_k) \\ \text{and } \textit{pseudo-constraint} & \hat{g} := \sum_k \hat{\alpha}_k g(x_k)\,. \end{array} \tag{3.7}$$

To help intuition, observe that

- pure cutting-planes imposes the constraint $\hat{g} = 0$: see (3.5);
- in a bundle method, this constraint is rather dualized (with multiplier $\hat{\lambda}$) and penalized (with penalty parameter $s/2$): see (3.6);
- if c is linear, then $\hat{c} = c(\hat{x})$: the pseudo-cost is the cost of the pseudo-schedule;
- if g is affine, then $\hat{g} = g(\hat{x})$: the pseudo-constraint is the constraint-value at the pseudo-schedule.

Observe also that $\hat{x}$ does not normally lie in X: it is not a "technically feasible" schedule, hence our terminology.

Remark 31. The solution of (3.4) is $\lambda_+ = \hat{\lambda} + s\hat{g}$, with $\hat{g}$ given in (3.7) and $\hat{\alpha}$ solving (3.6). Knowing that $\hat{g}$ is an "approximate subgradient" of θ at $\hat{\lambda}$, s appears as a stepsize and we see here that a bundle method resembles a subgradient method.

Our use of simplified notation (1.1), (1.2), instead of (2.5), (3.1), eliminates the additional dual variable μ. A more serious simplification is that, if (3.4) had positivity constraints, say $\max_{\mu \geqslant 0} \hat{\theta}(\mu) - \frac{\|\mu-\hat{\mu}\|^2}{2s}$, its dual would be fairly more intricate than (3.6). Deriving this dual would be possible, though. Calculations are left to a motivated reader, they follow [40], [9, §14.3]. □

Convergence of a bundle method means that

- the pseudo-cost tends to the dual optimal value: $\theta(\hat{\lambda}) \to \max \theta$ and $\hat{c} \to \max \theta$;
- the pseudo-constraint tends to 0: $\hat{g} \to 0$;
- besides, the stability center tends to a dual solution, if any: $\hat{\lambda} \to \lambda^*$.

Theory tells us that $\hat{c} \geqslant \max \theta$; besides, we have at each iteration

$$\theta(\lambda) \leqslant \theta(\hat{\lambda}) + \varepsilon + \hat{g}^\top(\lambda - \hat{\lambda}), \quad \text{for all } \lambda \in \mathbb{R}^m,$$

where we have set $\varepsilon := \hat{c} - \theta(\hat{\lambda}) \geqslant 0$. The algorithm can therefore be stopped when both ε and $\hat{g}$ are small: $\hat{\lambda}$ is then approximately dual optimal.

In practice, ε tends to 0 much faster than $\hat{g}$. Altogether, we see that the crucial point in a bundle method is to obtain a small $\hat{g}$, i.e. to obtain a pseudo-schedule which is approximately "pseudo-feasible" with respect to the linking constraints (more simply, it is approximately "feasible" if these constraints are affine).

3.4. Numerical illustration

In EdF's current implementation, the dual problem is solved by the bundle code described in [34], the restricted master problem being solved by K.C. Kiwiel's code QPDF4 described in [27, 28]. Some numerical illustrations following the same approach can be found in [32, 16, 15, 18] among others, see also [20] for most recent results; [37, 35] give comparative results with the interior-point and subgradient algorithms. Here we illustrate the behaviour of [34] on the model of §2.

We use three datasets, representative of various everyday situations, with different sets of available production units. In each of these sets, the primary reserve is neglected. There are 96 demand constraints, plus $2 \times 96 = 192$ reserve constraints, plus a number of daily constraints alluded to in Remark 21.

- Dataset A has 69 408 primal variables (power generation and reserves contributions for each unit at each time step) and 828 linking constraints (corresponding to 540 daily constraints); the maximal demand is 62 000 MW.
- Dataset B has 62 496 primal variables and 780 linking constraints (corresponding to 492 daily constraints); the maximal demand is 58 000 MW.
- Data C has 52 128 primal variables, 780 linking constraints, and a maximal demand of 52 000 MW.

Table 3.1. Statistics for three dual optimizations

Data	m	# iters	$\overline{\lvert B \rvert}$	$\overline{(R^2)_-}$	$\overline{(R^3)_-}$
A	828	200	6.70	5.59	2.18
B	780	189	2.21	2.99	0.0
C	780	174	1.70	2.86	0.0

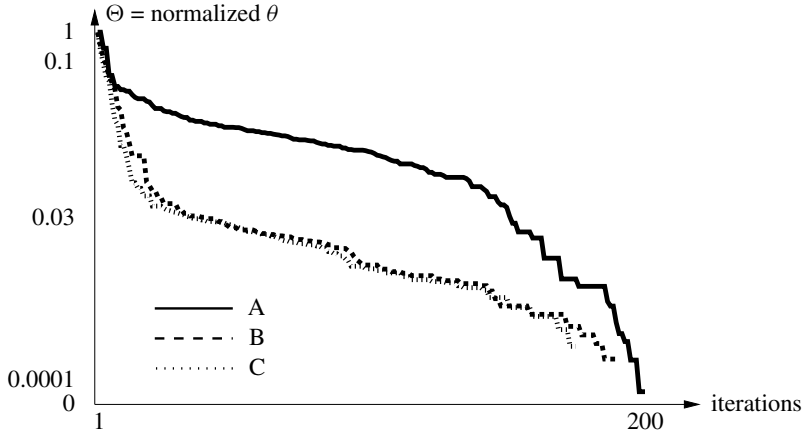

Fig. 3.2. Evolution of the (normalized) dual function

Table 3.1 gives the statistics: number of iterations, and constraints violations $\hat{g}$ by the final pseudo-schedule – see (3.7). An iteration is one resolution of the quadratic master, followed by one minimization of the Lagrangian. In the last three columns of Table 3.1, the notation B [resp. R^k] stands for the *real mismatch* in the balance constraint (2.3) [resp. the reserve constraints (2.4)], excluding the fictitious generators. In other words we have (recall that there is no primary reserve)

$$B_t = d_t^0 - \sum_{i \in \vartheta \cup \mathcal{H}} p_t^i \quad \text{and} \quad R_t^k = d_t^k - \sum_{i \in \vartheta \cup \mathcal{H}} R_t^{k,i}(p_t^i)\,,\ k = 2, 3\,. \tag{3.8}$$

An overline means average value (over time); the powers are given in MW.

It should be mentioned that excellent initializations are available for λ: between the initial and final iterates, the dual function increases by less than 5 %. Figure 3.2 gives an idea of the behaviour of $\theta(\hat{\lambda})$ along the iterations. For industrial privacy, θ-values are normalized: K being the total iteration number, the picture displays in logarithmic scale

$$\Theta_k := \frac{\theta(\hat{\lambda}_k) - \theta(\hat{\lambda}_K)}{\theta(\hat{\lambda}_1) - \theta(\hat{\lambda}_K)}\,, \quad \text{for } k = 1, \ldots, K-1\,,$$

a number varying from 1 to 0.

Due to the logarithmic scale, Fig. 3.2 is slightly deceiving, especially by the last iterations. Figure 3.3 zooms the values of Θ_k for dataset A, during the last 60 iterations.

As explained at the end of § 3.3, a crucial parameter in the algorithm is the pseudo-constraint $\hat{g}$ in (3.7). Figure 3.4 shows the evolution of $\|\hat{g}\|$ for the three examples, also in logarithmic scale.

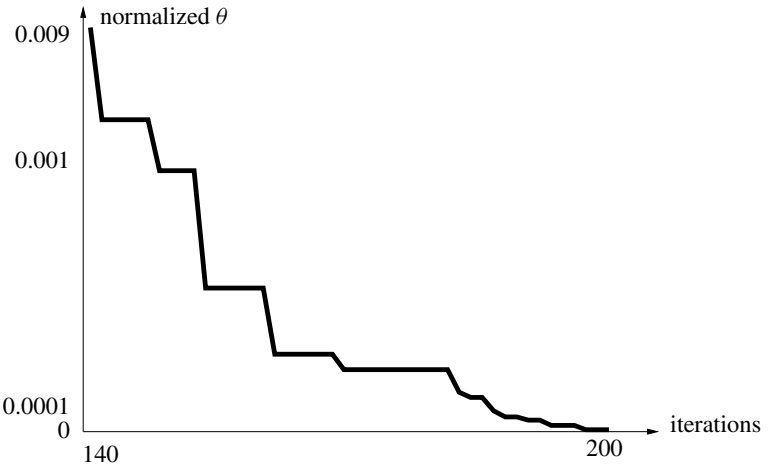

Fig. 3.3. Tail of the (normalized) dual function, dataset A

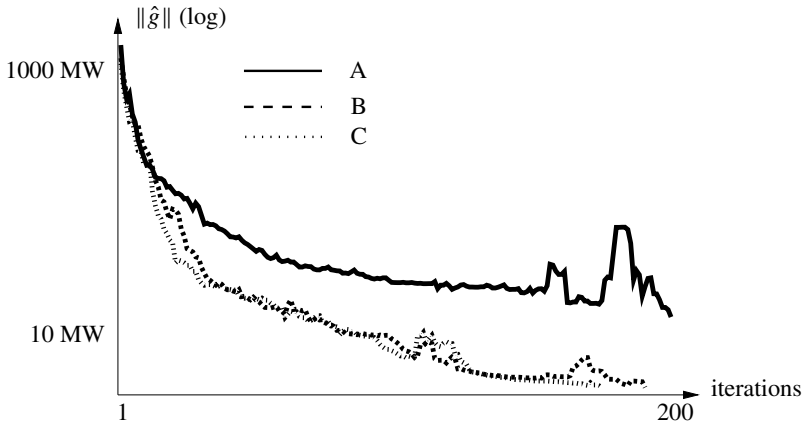

Fig. 3.4. Evolution of the pseudo-constraint

Remark 32. Passing from one iteration to the next in the bundle algorithm does not change much the restricted master (3.4) or (3.6):

- a new piece $L(x_+, \lambda)$ is introduced in $\hat{\theta}$ of (3.2) – resulting in an additional variable α_+ in (3.6),
- and/or the stepsize s is changed,
- and/or the stability center $\hat{\lambda}$ is updated.

Thus the restricted master lends itself to warm starts. However, the CPU time to solve (3.6) with QPDF4 is negligible compared with the time necessary for the local problems (3.1). In our reported experiments, the QP solver is actually used in a very primitive manner, where a cold start is performed at each iteration. This costs nothing in CPU and

can but improve robustness of the QP solver: propagation of roundoff errors is avoided in the Choleski matrix associated with (3.6). □

3.5. Reoptimization

A problem of raising importance in energy production is the following. Suppose (1.1) = (2.5) is solved, at least approximately. In particular the optimal cost is known, at least a good approximation of it (we will see in §5 that the dual bound *is* such a good approximation). Then comes a perturbation of the righthand side: d in (2.5) is replaced by $d + \delta d$, and one wishes to estimate *quickly* the corresponding perturbation of the optimal cost. A simple solution is to use duality: the sensitivity of the optimal cost (or of its dual approximation but this is good enough) with respect to the righthand side is λ^*. When the perturbation is large, however, this may result in substantial error; it is then safer to perform a reoptimization with the new righthand side.

Call $C(d)$ the dual optimal value of (2.5), considered as a function of the righthand side. Standard duality theory says that C is a convex function:

$$C(d + \delta d) \geqslant C(d) + \lambda^{*\top} \delta d \,.$$

Therefore $\lambda^{*\top} \delta d$ underestimates the change of the optimal cost, and is exact if *a posteriori*, λ^* is still a dual optimal solution for the new righthand side $d + \delta d$.

An obvious but important remark is that the righthand side has no influence whatsoever on the Lagrangian problems (3.1). Thus, each primal point x_k obtained during the first dual optimization would be obtained again during the second, if $L(x, \lambda_k)$ in (1.2) were minimized with the same λ_k. It follows that the function $\hat{\theta}(\lambda) + \lambda^\top \delta d$ – see (3.2) – is still a valid restricted dual function: it can be used right away for the second dual optimization, and this hot start may spare computing time.

We illustrate this technique with four typical perturbations of the demand d^0, given on Fig. 3.5. The data concern a Friday, with its dangling Saturday; the nominal demand oscillates between 37 000 and 54 000 MW; δd is a "stripe", extending between two time-steps – here 6am and 6pm – representing an additional need of some industrial customer. We make 4 scenarios, with stripes of width ranking from 1 000 to 4 000 MW.

Each of these perturbations does modify the optimal λ^*, as illustrated on Fig. 3.6.

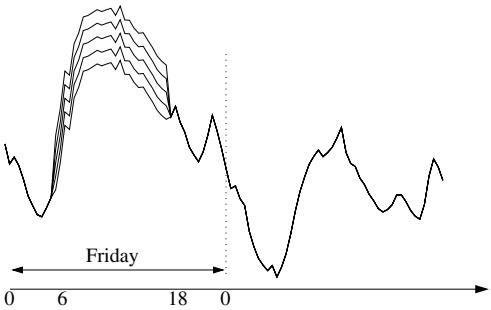

Fig. 3.5. Typical perturbation of the demand

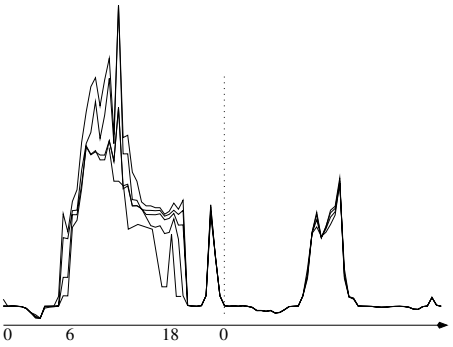

Fig. 3.6. Corresponding perturbation of λ^*

Table 3.2. Statistics for reoptimization

δd^0 (MW)	# iters	CPU (s)	δC	instead of
0	123	406	0	0
1 000	89	294	1.7	1.4
2 000	68	224	3.6	2.8
3 000	130	429	5.6	4.2
4 000	111	366	7.9	5.6

Table 3.2 gives the statistics of the re-optimization, comparing the actual and linearized increases of the cost; increases are given in fractions of the nominal $\max \theta$ (normalized, as in §3.4).

4. Primal recovery: two possible approaches

As a rule, the dual algorithm of §3 does not produce any primal solution, not even an approximately feasible one: recovering reasonable schedules calls for appropriate heuristics. The situation is then as follows: we have solved the dual problem associated with (1.1) and we have on hand

- a number of schedules x_k, obtained from (1.2) with $\lambda = \lambda_k$;
- the last stability center $\hat{\lambda}$, which solves (approximately) the dual problem;
- the last pseudo-schedule $\hat{x}$, which solves (approximately) a convexified form of (1.1),
- its accompanying pseudo-cost $\hat{c}$ and pseudo-constraint $\hat{g}$ of (3.7).

This data is available either from a standard column generation, or from a bundle algorithm[2].

Various approaches are possible for primal recovery:

(i) An exploration of the dual space around $\hat{\lambda}$ forces (1.2) to answer more feasible schedules; this is used for example in [48, 35]. Along the same lines,

[2] Incidentally, we also mention a lesser-known fact: $\hat{x}$ would also be available from a standard subgradient algorithm, see [1, 29] or [43, p. 116].

[3] uses a rather common technique in combinatorial optimization: to append redundant constraints, which might reduce the duality gap.

(ii) The primal space can also be explored directly. This is proposed in particular in [16, 15], where the exploration is made around $\hat{x}$.

(iii) Inspection of the x_k's reveals that many of their components take on the same value, for all k. Fixing these components drastically reduces the size of X, and thus allows a direct resolution of (1.1) with standard MIP tools; see [18, 44].

The design of heuristics at EdF is bound by the complexity of the model, in particular of the X_i's. The only practical schedules are those issued by (1.3) = (3.1) and it is hard to perturb them consistently. Pattern (i) is therefore followed, the Lagrangian being perturbed so that the dualized constraints will be better satisfied. The proposal of the present paper (§4.1 below) uses the pseudo-schedule $\hat{x}$ as in [16, 15], an approach which is probably a key to good heuristics. In §4.2 we present the heuristics currently used at EdF.

4.1. The primal proximal idea

It is crucial to understand – as in standard column generation – the particular role played by the pseudo-schedule $\hat{x}$. A simple example will illustrate it.

Example 41. Consider the problem with one variable and one constraint:

$$\min x\,, \quad x \in X := \{-1, 0, +1\}\,, \quad g(x) := x = 0\,,$$

with solution $x^* = 0$. Then $L(x, \lambda) = (1 + \lambda)x$ and $\theta(\lambda) = -|\lambda + 1|$. We may assume that the Lagrangian solver (1.2) computes

$$x(\lambda) = \begin{cases} 1 & \text{if } \lambda \leqslant -1\,, \\ -1 & \text{if } \lambda > -1\,, \end{cases} \quad \text{so that} \quad g(x_\lambda) = \begin{cases} 1 & \text{if } \lambda \leqslant -1\,, \\ -1 & \text{if } \lambda > -1\,, \end{cases}$$

the point being that only a very smart (1.2) could answer $x(-1) = 0$.

Start the dual algorithm at $\lambda_1 = 1$, thus obtaining $x_1 = -1$ (so that $\theta(\lambda_1) = -2$ and $\hat{\theta}(\lambda) = -1 - \lambda$). Then come a number of iterations depending on the particular algorithm chosen. We may assume for simplicity that we are using a bundle method with $s = 2$; with $\hat{\lambda} = 1$, we obtain $\lambda_2 = -1$. Then $x_2 = 1$ is computed, with $\theta(\lambda_2) = 0$. At this stage, we detect that λ_2 is dual optimal: in fact $\hat{\theta}(\lambda) = \min\{1 + \lambda, -1 - \lambda\}$ is maximal at $\lambda = -1$ with value $\hat{\theta}(-1) = 0 = \theta(-1)$. At the same time, (3.5) or (3.6) (with $\hat{\lambda} = 0$) provides $\hat{x} = \frac{1}{2}(x_1 + x_2) = 0$, the required primal solution. □

This naive example reveals the irrelevance of *all possible* answers x_k from the local solver (1.3): none of them, taken individually, is any good in terms of the dualized constraints. By contrast, their *convex combination* $\hat{x}$ appears as "better than the others". The remark can even be pushed further: X could be any set between $\{-1, +1\}$ and $[-1, +1]$, the situation would still be the same. All of the x_k's would look like random points, while $\hat{x}$ would somehow be the best possible point computable via duality.

Therefore, duality would probably give better primal results if the x_k's could somehow be made closer to $\hat{x}$. This suggests a simple idea: to add a quadratic term $r\|x-\hat{x}\|^2$ to the cost function $c(x)$, thus forcing the Lagrangian solver in (1.2) to answer better primal points, in terms of the dualized constraints. Of course, note that the perturbed dual function

$$\begin{aligned}\theta_r(\lambda) &:= \min_{x\in X} L_r(x,\lambda)\,, \text{ where}\\ L_r(x,\lambda) &:= c(x) + r\|x-\hat{x}\|^2 + \lambda^\top g(x)\end{aligned} \tag{4.1}$$

will no longer bound the primal cost from below. Besides, a dual alogrithm will produce a new pair $(\hat{x}_r, \hat{\lambda}_r)$, biased by $\hat{x}$.

When specialized to the unit-commitment problem (2.5), the local problems (3.1) become

$$\begin{aligned}&\min_{p^0\in\mathbb{R}^T} c^0(p^0) - \lambda^\top p^0 + r\|p^0-\hat{p}^0\|^2\,,\\ &\min_{p^{-k}\in\mathbb{R}^T_+} c^{-k}(p^{-k}) - \mu_k^\top p^{-k} + r\|p^{-k}-\hat{p}^{-k}\|^2 \qquad k=1,2,3\,,\\ &\min_{p^i\in P^i} c^i(p^i) - \lambda^\top p^i - \mu^\top R^i(p^i) + r\|p^i-\hat{p}^i\|^2 \qquad i\in\vartheta\,,\\ &\min_{w^i\in W^i} c^i(w^i) - \lambda^\top p^i(w^i) - \mu^\top R^i(p^i(w^i)) + r\|w^i-\hat{w}^i\|^2 \quad i\in\mathcal{H}\,.\end{aligned} \tag{4.2}$$

The decomposability property is preserved, only the form of the cost has changed; dynamic programming ($i\in\vartheta$) is not affected, and interior-point ($i\in\mathcal{H}$) accommodates the quadratic term easily; as for the fictitious problems, they are trivial anyway.

Remark 42. Alternatively, one can penalize the deviation of the (local) *constraint values*. With the schematic form (1.1), (1.3), the penalized local problems become

$$\min_{x^i\in X^i} c^i(x^i) + \lambda^\top g^i(x^i) + r\|g^i(x^i) - g^i(\hat{x}^i)\|^2\,.$$

This is probably more sensible, as it gets closer to the augmented Lagrangian idea of §4.2 below. Note also how this is reminiscent of primal decomposition [17].

With the specific problem (3.1), we obtain

$$\min_{p^i\in P^i} c^i(p^i) - \lambda^\top p^i - \mu^\top R^i(p^i) + r\|p^i-\hat{p}^i\|^2 + r\|R^i(p^i) - R^i(\hat{p}^i)\|^2\,,\quad i\in\vartheta$$

for the thermal units, and likewise for the others. However, a difficulty appears for the hydro-valleys: since the production is a nonlinear function of the control variable, the quadratic term $r\|p^i(w^i) - p^i(\hat{w}^i)\|^2$ results in a nonconvex quadratic program, which can no longer be conveniently solved.

In what follows, we will content ourselves with the formulation (4.2), which represents a handy notation for the actual implementation. □

The schedules computed in (4.2) are *technically feasible* and *easy to compute*; due to their tendency to be closer to the pseudo-schedule, the hope is that some of them are good in terms of satisfaction of the dualized constraints. In summary, the proposed algorithm to solve (2.5) is composed of two phases:

Algorithm 43 (phase 1 - phase 2 for unit-commitment)**.**

PHASE 1. *Solve (by a bundle method) the dual problem associated with* (3.1).
Obtain a dual solution $\hat{\lambda}$ *and a pseudo-schedule* $\hat{x}$.
Choose a penalty term $r > 0$.

PHASE 2. *Solve (by a bundle method) the dual problem associated with* (4.2).
During this resolution, record the best schedule computed in (4.2). □

Best schedules are of course those giving the best real + fictitious cost (remember Remark 22). They can also be recorded during Phase 1. A reasonable value for r can be computed: it must simply balance the primal costs and the (squared) constraint violations obtained during the first phase. The perturbed dual problem is again convex, so the dual algorithm can be started anywhere in Phase 2 – for example on $\hat{\lambda}$. However, in contrast with §3.5, the old x_k's are of no use to restart the second phase.

It is interesting to relate Algorithm 43 to the so-called (primal) *proximal algorithm*. Indeed, for given $\hat{x} \in \mathbb{R}^n$, consider the following perturbation, "more convex" than (1.1):

$$\min c(x) + r\|x - \hat{x}\|^2\,, \quad x \in X\,, \quad g(x) = 0\,, \tag{4.3}$$

whose optimal solution (assumed unique for simplicity) is called the *proximal point* of $\hat{x}$; we will denote it by $p_r(\hat{x})$. Clearly, if $\hat{x}$ solves (1.1) then $p_r(\hat{x}) = \hat{x}$. The converse is also true under general assumptions: [13, Thm 1.4]. To solve (1.1) can therefore be viewed as to find a fixed point of the mapping p_r. This is an incentive for the proximal algorithm:

$$\begin{array}{l}\text{Start from some } \hat{x}_1\text{; repeat}\\ \hat{x}_{\ell+1} = p_r(\hat{x}_\ell)\,,\end{array} \tag{4.4}$$

designed in the 60's to minimize ill-condtioned quadratic functions: [5].

Interestingly, the idea was used in [6] for nonconvex problems. More precisely, assume that (1.1) has C^2 data, and that standard second-order conditions are satisfied. Then, for r large enough, the proximal algorithm (4.4) converges to a local minimum x^* of (1.1) *providing that* its initialization $\hat{x}_1$ is close enough to x^*. Note from the stated assumptions that the perturbed Lagrangian $L_r(\cdot, \lambda)$ of (4.1) becomes convex and there is no duality gap: the proximal points can be computed by Lagrangian relaxation.

Here, the assumptions of [6] are far from satisfied. However we retain that a good initialization is important. From this point of view, Phase 2 of Algorithm 43 can be viewed as the first iteration of the proximal algorithm, *initialized on the pseudo-schedule* computed by Phase 1: $\hat{x}_1 = \hat{x}$. Phase 2 terminates with some $\hat{x}_2$ (probably different from $\hat{x}$) and can be repeated if necessary. We have not implemented this possilibity: in our experiences, one execution of Phase 2 has always been sufficient.

4.2. An approach via augmented Lagrangian

A known technique to cancel the duality gap is the augmented Lagrangian [23, 38, 40, 41, 7]. Consider the equivalent form of (1.1)

$$\min c(x) + \pi\|g(x)\|^2\,, \quad x \in X\,, \quad g(x) = 0\,. \tag{4.5}$$

The corresponding dual function is

$$\theta^\pi(\lambda) := \min_{x\in X} L^\pi(x,\lambda)\,, \qquad \text{where}$$
$$L^\pi(x,\lambda) := c(x) + \lambda^\top g(x) + \pi\|g(x)\|^2\,.$$

Under mild assumptions (certainly satisfied by the unit-commitment problem, see [24, Thm. XII.5.2.2]), there is no duality gap: for π large enough, $\sup\theta^\pi$ is the optimal value of (1.1). Unfortunately, the method is only theoretical, as $L^\pi(\cdot,\lambda)$ is "impossible" to minimize: cross-products of the type $x^i x^j$ destroy the decomposability property.

Remark 44. When (1.1) is convex, the above augmented dual function satisfies

$$\theta^\pi(\hat\lambda) = \max_{\lambda\in\mathbb{R}^m} \theta(\lambda) - \frac{1}{4\pi}\|\lambda-\hat\lambda\|^2$$

(see [25, Prop. XII.5.2.3]). Comparing with (4.3), a dual method using augmented Lagrangian appears as a *dual* proximal algorithm to maximize the dual function θ: the proximal term $\|\lambda-\hat\lambda\|^2$ occurs in the dual space. Compare also with the bundle approach (3.4). □

The method currently in operations at EdF is based on augmented Lagrangian; it is made implementable via a mechanism due to [12], which works as follows; see [39]. First linearize the trouble-making term $\|g(x)\|^2$; denoting by $\tilde x$ the linearization point, the augmented Lagrangian becomes

$$\tilde L^\pi(x,\lambda,\tilde x) = c(x) + \lambda^\top g(x) + 2\pi g(\tilde x)\frac{\partial g}{\partial x}(\tilde x)(x-\tilde x)\,.$$

Then the local problems can cope with this Lagrangian, to compute the iterates x_k. However, the whole beneficial effect of augmented Lagrangian heavily relies on a *quadratic* term (to stabilize the x_k's); $\tilde L^\pi$ is therefore modified further to the "augmented-linearized-proximized" Lagrangian

$$\begin{aligned}\tilde L^{r\pi}(x,\lambda,\tilde x) &= \tilde L^\pi(x,\lambda,\tilde x) + r\|x-\tilde x\|^2\\ &= c(x) + \lambda^\top g(x) + 2\pi g(\tilde x)\tfrac{\partial g}{\partial x}(\tilde x)(x-\tilde x) + r\|x-\tilde x\|^2\,.\end{aligned}$$

Its minimization (for given $\lambda=\lambda_k$ and $\tilde x=\tilde x_k$) by the local problems gives the iterate x_k. Then λ is updated according to the standard scheme in augmented Lagrangian (see [7, §4.2] for example)

$$\lambda_{k+1} = \lambda_k + \pi\frac{\partial\tilde L^{r\pi}}{\partial\lambda}(x_k,\lambda_k,\tilde x_k) = \lambda_k + \pi g(x_k)\,.$$

As for $\tilde x$, it is simply set to the current iterate: $\tilde x_{k+1} = x_k$.

The quadratic term $\|x-\tilde x\|^2$ introduces the proximal mechanism; the above approach resembles Algorithm 43, but with a number of differences:
– the insertion of the linear term $2\pi g(\tilde x)\frac{\partial g}{\partial x}(\tilde x)x$,

– a "diagonalized" update of λ and of the primal stability center: here they are updated simultaneously, a strategy *à la* Arrow-Hurwicz [2]; by contrast, Algorithm 43 uses a strategy *à la* Uzawa [46], updating $\hat{x}$ (or x_ℓ, to follow the notation of (4.4)) only after having performed a full optimization on λ.

EdF's implementation has two more differences:

– the stability center $\tilde{x}_0$ is initialized on the last x_k computed during Pase I (but our example at the beginning of §4 suggests that this is not a sound idea),

– the approach is actually applied to a Lagrangian decomposition form of (2.5), with a duplication of the primal variables (see [21, 22] and [33, §4.3]):

$$\min c(x)\,, \quad x \in X \subset \mathbb{R}^n\,, \quad g(y) = 0 \in \mathbb{R}^m\,, \quad x = y$$

and it is the constraints $x = y$ that are dualized.

5. Primal recovery: comparative illustrations

The two techniques described described in §4 – namely PS (pseudo-schedule) of §4.1 and AL (augmented Lagrangian) of §4.2 – have been extensively compared on real instances. Invariably the comparison turned to the advantage of PS, mainly in terms of robustness. The present section illustrates on a few concrete examples the typical results observed.

5.1. Standard situations

We start our comparison with the three datasets of §3.4. For each example, we compare in Table 5.1 the schedules obtained, in terms of their cost and actual constraint violations. The column "cost" records the excess of the total cost (real + fictitious) over the lower bound $\max \theta$ obtained in Phase 1; and see (3.8) for the notation B, R^k.

The table also shows the iteration number where the best schedule has been obtained, and the number of iterations until termination (remember that AL has no reliable stopping criterion). An important advantage of PS is to find its best solution before AL – in addition to the availability of a stopping criterion.

5.2. A difficult case

In Case D, used in this section, the power load reaches a peak at 70 000 MW, thus providing a suitable test to illustrate the robustness of PS. Phase 2 becomes substantially

Table 5.1. Statistics for three standard examples

Data	Alg.	cost	$\overline{\lvert B\rvert}$	$\overline{(R^2)\_}$	$\overline{(R^3)\_}$	Iteration #	
		%		megawatts		best	stop
A	AL	1.27	44.0	12.7	0.8	155	200
	PS	0.56	36.4	3.7	0.4	28	65
B	AL	0.22	22.5	6.1	0.0	95	200
	PS	0.14	21.6	6.4	0.0	8	33
C	AL	0.26	17.6	11.8	0.0	163	200
	PS	0.16	29.7	10.9	0.0	11	36

harder to solve, as is suggested by the iteration numbers reported in Table 5.2; note also how expensive is the schedule found by AL (8.1% more than the lower bound). On the other hand, the average mismatches of the linking constraints seem acceptable. However, the maximal shortage in power, reached at a certain time $\bar{t}$, is unacceptable for AL (1 659 MW – remember from (3.8) $B < 0$ means an insufficient production; this schedule is totally useless) and very bad for PS. We note for future use that there is also a shortage of tertiary reserve at $\bar{t}$.

Obviously, both algorithms should have started an additional unit at $\bar{t}$, but they failed to find such a solution. Actually, the blame is on the fictitious generator: at $\bar{t}$, the magnitude of λ becomes comparable to the slopes of c^0 (see Fig. 2.2). Then the algorithms use an unduly large amount of fictitious power: this is cheaper than starting an additional unit (probably most expensive, otherwise it would already be on!), and keeping it on for a possibly long time (because of technological constraints).

Table 5.3 therefore reports on an experiment with a more brutal and simple $c_t^0(x) := 10^3|x|$. Because c^0 can hardly be given a sensible economic interpretation, the table also records real costs, excluding the fictitious ones. Observe the reasonable discrepancies for PS, which proposes a cheaper schedule than AL. Actually, the latter again failed to converge. Note in particular that there is now an (unacceptable) *overproduction* at $\bar{t}$; and it is accompanied by an *underproduction* of tertiary reserve; this is nonsense: shifting some of the power to the tertiary reserve would do only good.

5.3. Inserting primary reserve

Finally, two cases – E and F – consider the three reserves altogether. Table 5.4 gives the statistics, as before. Only real costs are considered, the cost of AL being normalized to 1. Again, PS finds faster more economic schedules.

As a general conclusion to all of these experiments, we summarize the advantages of the PS algorithm of §4.1.

– In normal situations, the best solution is generally found earlier in the iterative process.
– It has an implementable stopping criterion; and it can be iteratively repeated if necessary.
– Its solutions are statistically cheaper than the solutions obtained with the AL algorithm.
– Mismatches of the linking constraints are smaller.

Table 5.2. Case D: an unusual peakload

Alg.	cost	$\overline{\lvert B\rvert}$	$\overline{(R^2)_-}$	$\overline{(R^3)_-}$	$B_{\bar{t}}$	$R^3_{\bar{t}}$	Iter. #	
	%	megawatts			megawatts		best	stop
AL	8.1	118.1	3.8	2.0	-1659	-5	200	200
PS	1.5	105.6	6.7	2.3	-786	-12	195	200

Table 5.3. Case D with increased c^0

Alg.	cost	cost	$\overline{\lvert B\rvert}$	$\overline{(R^2)_-}$	$\overline{(R^3)_-}$	$B_{\bar{t}}$	$R^3_{\bar{t}}$	Iter. #	
	%	(real)	megawatts			megawatts		best	stop
AL	33	-0.21	79.2	15.2	50.4	1301	-727	200	200
PS	12	-1.14	31.6	18.8	8.1	-42	-56	8	200

Table 5.4. Two cases with primary reserve

Data	Alg.	cost	$\overline{\lvert B\rvert}$	$\overline{(R^1)\_}$	$\overline{(R^2)\_}$	$\overline{(R^3)\_}$	best iter
			megawatts				
E	AL	1	45.7	0.	15.3	0.	199
	PS	0.98	29.8	0.2	9.0	0.	37
F	AL	1	47.5	2.2	9.0	0.3	198
	PS	0.99	46.7	1.5	9.0	0.1	72

– It is easier to tune and more robust, especially with respect to the fictitious cost.
– It is less sensitive to the addition of new linking constraints.
– More generally, it only uses standard duality, which makes it more satisfactory from a theoretical point of view.

References

1. Anstreicher, K., Wolsey, L.A.: On dual solutions in subgradient optimization. Unpublished manuscript, CORE, Louvain-la-Neuve, Belgium, 1993
2. Arrow, K.J., Hurwicz, L.: Reduction of constrained maxima to saddle point problems. In: Neyman, J. (ed.) Proceedings of the 3rd Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1956, pp. 1–26
3. Baldick, J.: The generalized unit commitment problem. IEEE Transactions on Power Systems **10** (1), 465–475 (1995)
4. Batut, J., Renaud, A.: Daily generation scheduling with transmission constraints: a new class of algorithms. IEEE Transactions on Power Systems **7** (3), 982–989 (1992)
5. Bellman, R., Kalaba, R., Lockett, J.: Numerical Inversion of the Laplace Transform. Elsevier, 1966
6. Bertsekas, D.P.: Convexification procedures and decomposition methods for nonconvex optimization problems. Journal of Optimization Theory and Applications **29**, 169–197 (1979)
7. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, 1995
8. Bertsekas, D.P., Lauer, G.S., Sandell, N.R., Posberg, T.A.: Optimal short-term scheduling of large-scale power systems. IEEE Transactions on Automatic Control **AC-28**, 1–11 (1983)
9. Bonnans, J.F., Gilbert, J.Ch., Lemaréchal, C., Sagastizábal, C.: Numerical Optimization. Springer Verlag, 2003
10. Cheney, E., Goldstein, A.: Newton's method for convex programming and Tchebycheff approximations. Numer. Math. **1**, 253–268 (1959)
11. CIGRE SC 38. Task Force 38-04-01. Unit commitment, final report, 1997
12. Cohen, G.: Auxiliary problem principle and decomposition of optimization problems. J. Optim. Theory Appl. **32**, 277–305 (1980)
13. Daniilidis, A., Lemaréchal, C.: On a primal-proximal heuristic in discrete optimization. Mathematical Programming **104**, 105–128 (2005). Also available as: Inria Research Report 4550, www.inria.fr/rrrt/rr-4550.html.
14. Falk, J.E.: Lagrange multipliers and nonconvex programs. SIAM J. Cont. **7** (4), 534–545 (1969)
15. Feltenmark, S., Kiwiel, K.C.: Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems. SIAM J. Optim. **10** (3), 697–721 (2000)
16. Feltenmark, S., Kiwiel, K.C., Lindberg, P.O.: Solving unit commitment problems in power production planning. In: U. Zimmermann (ed.) Operations Research Proceedings **1996**, 236–241 (1997)
17. Geoffrion, A.M.: Primal resource-directive approaches for optimizing nonlinear decomposable systems. Operations Research **18** (3), 375–403 (1970)
18. Gollmer, R., Nowak, M.P., Römisch, W., Schultz, R.: Unit commitment in power generation – a basic model and some extensions. Annals of Operations Research **96**, 167–189 (2000)
19. Gonzalez, R., Bongrain, M.P., Renaud, A.: Unit commitment handling transmission constraints with an interior point method. In Proceedings of the 13th PSCC conference, Trondheim, **2**, 715–723 (1999)
20. Gröwe-Kuska, N., Römisch, W.: Stochastic unit commitment in hydro-thermal power production planning. In: S.W. Wallace, W.T. Ziemba (eds.), Applications of Stochastic Programming, Series in Optimization. SIAM Publications, MPS–SIAM, 2004
21. Guignard, M., Kim, S.: Lagrangean decomposition: a model yielding stronger Lagrangean bounds. Mathematical Programming **39** (2), 215–228 (1987)
22. Guignard, M., Kim, S.: Lagrangean decomposition for integer programming: theory and applications. RAIRO Recherche Opérationnelle **21** (4), 307–323 (1987)

23. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**, 303–320 (1969)
24. Hiriart-Urruty, J.-B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Springer Verlag, Heidelberg, 1993. Two volumes
25. Hiriart-Urruty, J.-B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Springer Verlag, Heidelberg, 1993. Two volumes
26. Kelley, J.E.: The cutting plane method for solving convex programs. J. Soc. Indust. Appl. Math. **8**, 703–712 (1960)
27. Kiwiel, K.C.: A dual method for certain positive semidefinite quadratic programming problems. SIAM Journal on Scientific and Statistical Computing **10** (1), 175–186 (1989)
28. Kiwiel, K.C.: A Cholesky dual method for proximal piecewise linear programming. Numer. Math. **68**, 325–340 (1994)
29. Larsson, T., Patriksson, M., Strömberg, A.B.: Ergodic, primal convergence in dual subgradient schemes for convex programming. Mathematical Programming **86** (2), 283–312 (1999)
30. Lemaréchal, C.: Lagrangian relaxation. In: M. Jünger, D. Naddef (eds.), Computational Combinatorial Optimization, Springer Verlag, Heidelberg, 2001, pp. 115–160
31. Lemaréchal, C.: The omnipresence of Lagrange. 4OR, **1** (1), 7–25 (2003)
32. Lemaréchal, C., Pellegrino, F., Renaud, A., Sagastizábal, C.: Bundle methods applied to the unit-commitment problem. In: J. Doležal, J. Fidler (eds.), System Modelling and Optimization, Chapman and Hall, 1996, pp. 395–402
33. Lemaréchal, C., Renaud, A.: A geometric study of duality gaps, with applications. Mathematical Programming **90** (3), 399–427 (2001)
34. Lemaréchal, C., Sagastizábal, C.: Variable metric bundle methods: from conceptual to implementable forms. Mathematical Programming **76** (3), 393–410 (1997)
35. Madrigal, M., Quintana, V.H.: An interior-point/cutting-plane method to solve unit commitment problems. IEEE Transactions on Power Systems **15** (3), 1022–1027 (2000)
36. Magnanti, T.L., Shapiro, J.F., Wagner, M.H.: Generalized linear programming solves the dual. Management Science **22** (11), 1195–1203 (1976)
37. Pellegrino, F., Renaud, A., Socroun, T.: Bundle method and augmented Lagrangian methods for short-term unit commitment. In Proceedings of the 12th PSCC Conference, Dresden **2**, 730–739 (1996)
38. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: R. Fletcher (ed.), Optimization. Academic Press, London, New York, 1969
39. Renaud, A.: Daily generation management at Electricité de France: from planning towards real time. IEEE Transactions on Automatic Control **38** (7), 1080–1093 (1993)
40. Rockafellar, R.T.: The multiplier method of Hestenes and Powell applied to convex programming. J. Optim. Theory Appl. **6**, 555–562 (1973)
41. Rockafellar, R.T.: Augmented Lagrange multiplier functions and duality in nonconvex programming. SIAM J. Cont. **12**, 268–285 (1974)
42. Sheblé, G.B., Fahd, G.N.: Unit commitment literature synopsis. IEEE Transactions on Power Systems **9**, 128–135 (1994)
43. Shor, N.Z.: Minimization methods for non-differentiable functions. Springer Verlag, Berlin, 1985
44. Takriti, S., Birge, J.R.: Using integer programming to refine lagrangian-based unit commitment solutions. IEEE Transactions on Power Systems **15** (1), 151–156 (2000)
45. Terlaky, T. (ed.): Interior Point Methods of Mathematical Programming. Kluwer Academic Press, Dordrecht, 1996
46. Uzawa, H.: Iterative methods for concave programming. In: K. Arrow, L. Hurwicz, H. Uzawa (eds.), Studies in Linear and Nonlinear Programming. Stanford University Press 1959, pp 154–165
47. Vanderbeck, F.: A generic view at the Dantzig-Wolfe decomposition approach in mixed integer programming: paving the way for a generic code. Working paper U0222, University of Bordeaux, Talence, France, 2002
48. Zhuang, F., Galiana, F.D.: Towards a more rigorous and practical unit commitment by lagrangian relaxation. IEEE Transactions on Power Systems **3** (2), 763–773 (1988)