Solving the Unit Commitment Problem of Hydropower Plants Via Lagrangian Relaxation and Sequential Quadratic Programming✦

E. C. FINARDI✉ erlon@labplan.ufsc.br
E. L. da SILVA edson@labplan.ufsc.br
LabPlan – Laboratório de Planejamento de Sistemas de Energia Elétrica. Campus Universitario, Trindade, Florianópolis 88040-900, SC, Brasil.

C. A. SAGASTIZABAL sagastiz@impa.br
IMPA – Instituto de Matematica Pura e Aplicada. Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro 22460-320, RJ, Brasil.

Abstract: We consider the optimal scheduling of hydropower plants in a hydrothermal interconnected system. This problem, of outmost importance for large-scale power systems with a high proportion of hydraulic generation, requires a detailed description of the so-called hydro unit production function. In our model, we relate the amount of generated hydropower to nonlinear tailrace levels; we also take into account hydraulic losses, turbine-generator efficiencies, as well as multiple 0-1 states associated with forbidden operation zones. Forbidden zones are crucial to avoid nasty phenomena such as mechanical vibrations in the turbine, cavitation, and low efficiency levels. The minimization of operating costs subject to such detailed constraints results in a large-scale mixed-integer nonlinear programming problem. By means of Lagrangian Relaxation, the original problem is split into a sequence of smaller and easy-to-solve subproblems, coordinated by a dual master program. In order to deal better with the combinatorial aspect introduced by the forbidden zones, we derive three different decomposition strategies, applicable to various configurations of hydro plants (with few or many units, which can be identical or different). We use a Sequential Quadratic Programming algorithm to solve nonlinear subproblems. We assess our approach on a real-life hydroelectric configuration extracted from the south sub region of the Brazilian hydrothermal power system.

Keywords: Hydrothermal Systems, Unit Commitment Problems, Lagrangian Relaxation, Sequential Quadratic Programming.

1. Introduction

The optimal generation scheduling is an important daily activity for electric power generation companies. The goal is to determine which units are to be used in order to generate enough power to satisfy demand requirements and various technological constraints, with minimum operating cost. In particular, hydrothermal systems must consider the stream-flow equations for reservoirs. These equations couple all the reservoir along a hydro-valley, because the amount of outflow water[1] released by one power plant affects water volumes in all the plants downstream. Furthermore, water travel times and alternative uses of water, such as irrigation or flood control, for example, must also be taken into account.

The optimal scheduling of hydropower plants is called the Hydro Unit Commitment (HUC) problem. To solve the HUC problem, a highly sophisticated modeling for the operation of hydro plants is required. Specifically, a hydropower plant may be composed of several turbine-generator groups, referred to in this work as "units". The amount of power generated by one hydro unit depends on the efficiency of both the turbine and the generator, as well as on the net head and the unit turbined outflow. In turn, the net head is a nonlinear function of the storage and of the reservoir outflow. The joint turbine-generator efficiency varies with the water net head and the unit turbined outflow. In addition, the existence of forbidden operation regions prevents the unit from generating power in a wide and continuous range. These regions, modeled by 0-1 variables, aim at avoiding vibrating modes that may produce unwanted power oscillations, cavitation phenomena, and low levels of efficiency. Thermal power plants have simpler production functions, but they need start-up and shut-down times and they often present nonlinear operating costs.

As a result, the hydrothermal unit commitment problem is a large-scale mixed-integer nonlinear programming problem which can only be effectively solved by applying decomposition techniques. Lagrangian Relaxation (LR) is particularly suitable for this type of problems, [1-5], although some other methodologies

---

✦ Research supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Centro de Pesquisas de Energia Elétrica (CEPEL). The authors acknowledge with grateful thanks the contributions made by Maria Elvira Maceira and Albert Cordeiro Geber de Melo from CEPEL.
✉ Corresponding author.
1 Turbined and spilt water flow.

have also been proposed [6]. However, so far none of the works in the area has considered a modeling as comprehensive as ours, with representation of hydraulic losses, nonlinear tailrace levels, turbine-generator efficiencies, and forbidden operation zones.

With respect to the solution method, our contribution consists in a thorough analysis of three different decomposition schemes, all derived from LR. The first strategy relies on a complete enumeration of all possible 0-1 operating states of the units composing a hydropower plant. This approach is suitable for plants with a low number of identical units. The second strategy, requiring less computation effort, is applicable for plants with many units that are different and have many forbidden regions. The third strategy combines the two other approaches, and can be used in systems with both types of hydro plants.

Many of the subproblems resulting from our decomposition schemes are Nonlinear Programs (NLP) of small size. We solve them by a Sequential Quadratic Programming (SQP) method [7,8], in a quasi-Newton variant [9,10], which presents good convergence properties.

Our work is organized as follows. In Section 2 we give the hydro plants and units modeling. Section 3 is devoted to the mathematical formulation of the optimal scheduling problem. The solution strategy, with the three decomposition schemes, is given in Section 4. In Section 5 we report numerical results on a hydrothermal system corresponding to Brazil's southern electric sub region. We end in Section 6 with some concluding remarks.

2. Hydro Generating Units

For a unit j, the generated power, ph_j, expressed in [MW], depends on the unit turbined outflow, q_j, on the net water head, hl_j, and on joint turbine and generator efficiency, η_j:

$$ph_j = 9{,}81 \times 10^{-3} \eta_j hl_j q_j \tag{1}$$

The net water head has the expression:

$$hl_j = fcm - fcj(Q,s) - k_j q_j^2 \tag{2}$$

Here, fcm stands for the forebay level. For short term horizon problems, as in our case, the forebay level remains practically constant, specially in the Brazilian case, whose huge reservoirs have typical regularization levels of a couple of years. Therefore, we consider fcm constant. By contrast, the downstream level $fcj(.)$ varies abruptly in short times, mainly due to the plant turbined outflow, Q, given by the addition of the outflows of all the units composing the plant. For some power plant configurations, fcj also varies with the reservoir spillage, s. In (2), the term $k_j q_j^2$ represents hydraulic losses resulting from friction of the water in penstock, with k_j a constant in s^2/m^5 [11].

The unit efficiency, depending on hl_j and q_j, is usually represented by hill diagrams given by the factory; see Figure 1. We estimate it by interpolation; see [12], using a polynomial function:

$$\eta_j = \rho_{0j} + \rho_{1j} q_j + \rho_{2j} hl_j + \rho_{3j} hl_j q_j + \rho_{4j} q_j^2 + \rho_{5j} hl_j^2 \tag{3}$$

where the coefficients $\rho_{0j},\ldots,\rho_{5j}$ have been computed beforehand.

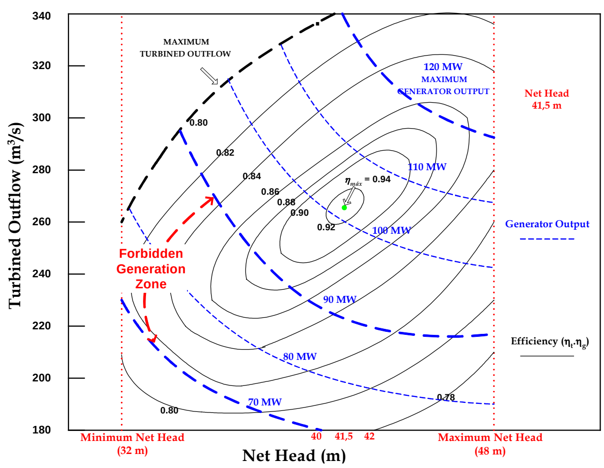

Figure 1 – Hill Diagram.

Figure 1 also displays some important operating constraints on the turbine-generator group. For example, for net head values smaller than the nominal level (41,5 m), the turbine is unable to make the generator attain its nominal power (120 MW). On the other hand, for values higher than 41,5 m, the power limit is given by the generator capabilities, because the turbine could effectively reach power levels beyond 120 MW. Since at some power levels cavitation phenomena and nasty mechanic vibrations may appear, in order to extend the lifetime of the unit and to avoid power oscillations, such power levels are forbidden. For example, Figure 1 shows a forbidden operation region ranging from 70 to 90 MW.

By combining (1)–(3) we obtain our model for the hydro production function:

$$ph_j = 9{,}81\times 10^{-3}(\rho_{0j} + \rho_{1j}q_j + \rho_{2j}hl_j + \rho_{3j}hl_jq_j + \rho_{4j}q_j^2 + \rho_{5j}hl_j^2)hl_jq_j \quad (4)$$

For the Brazilian case, *fcj*(Q,s) is represented by a fourth degree polynomial. Therefore, from (4) we see that ph_j is a polynomial[2] of degree 12 on the variables Q and s, and of order 7 in the variable q_j.

At first sight our model may appear as "too complicated"; however, it is important to realize that only such a detailed description can accurately represent the diverse amounts of power generated by a unit at different operating states.

3. Problem Formulation

The objective function for the thermal-HUC problem has the expression:

$$\sum_{t=1}^{T}\sum_{i=1}^{I} c_{it}(pt_{it}) + \alpha \quad (5)$$

Here, the planning horizon is composed by T time steps, the thermal mix has I plants, $c_{it}(.)$ represents the operating cost of the *i-th* thermal plant at time step t, and α stands for the system expected future cost at the end of the planning horizon; see (8) below. Frequently $c_{it}(.)$ includes fixed costs as well as fuel costs related to start-up and nominal generation of thermal units [2],[5].

We formulate the thermal-HUC constraints by splitting them into three different subsets, C_H, C_T and C_{HT}, corresponding to the respective variables involved namely, hydraulic, thermal, or both. Each subset is characterized by a specific type of coupling, such as units in the same power plant along different time steps (time coupling), or different power plants in a given time step (space coupling).

We now proceed to give each constraint in detail.

4.1 *Constraints involving only hydraulic variables* (C_H)

- Stream-flow balance equation:

$$v_{r,t+1} + Q_{rt} + s_{rt} - \sum_{m\in\Re_+^{(r)}} (Q_{m,t-\tau_{mr}} + s_{m,t-\tau_{mr}}) - v_{rt} = y_{rt} \quad (6)$$

We use the index r for reservoirs, v is the reservoir storage, y is the incremental inflow, $\Re_+^{(r)}$ is a set gathering all reservoirs upstream the *r-th*, and τ_{mr} is the water travel time between reservoirs m and r.

- Maximum v_r^{max} and minimum v_r^{min} storage, and maximum spillage s_r^{max} per reservoir:

$$v_r^{\min} \le v_{r,t+1} \le v_r^{\max} \qquad 0 \le s_{rt} \le s_r^{\max} \quad (7)$$

- Expected future cost function, given by longer term planning models, and estimating the cost of using today water that might become necessary (and expensive) in the future; see [13]. It is a piecewise affine function that depends on the final levels of stocked water, v_{rT}:

$$\alpha = f(v_{rT}) \quad (8)$$

- Penstock water balance equation per reservoir:

$$Q_{rt} = \sum_{j=1}^{J(r)} q_{jrt} \quad (9)$$

$J(r)$ is the number of generating units in reservoir r.

---

[2] Since $hl_j=f(q_j^2,Q^4,s^4)$, it follows that $\eta_j=f(q_j^4,Q^8,s^8)$ and, thus, by (1), $ph_j=f(q_j^7,Q^{12},s^{12})$.

- Power limits, given for each operating region of the unit:

$$\sum_{k=1}^{\Phi_{jr}} ph_{jkrt}^{mín} z_{jkrt} \leq ph_{jrt}(q_{jrt}, Q_{rt}, s_{rt}) \leq \sum_{k=1}^{\Phi_{jr}} ph_{jkrt}^{máx} z_{jkrt} \tag{10}$$

Φ_{jr} denotes the total number of non-forbidden regions of the *j-th* unit in reservoir *r*; *k* is the corresponding index, and $ph_{jkrt}^{min,max}$ stand for the minimum and maximum power limits. The binary variable z_{jkrt} is 1 if the *j-th* unit in reservoir *r* is operating in the *k-th* region at time step *t*, and it is set to 0 otherwise.

- Reservoir power balance:

$$PH_{rt} = \sum_{j=1}^{J(r)} ph_{jrt}(q_{jrt}, Q_{rt}, s_{rt}) \tag{11}$$

- Reserve constraints:

$$\left(\sum_{j=1}^{J(r)} \sum_{k=1}^{\Phi_{jr}} ph_{j1rt}^{máx} z_{jkrt} - PH_{rt} \right) \geq rh_{rt} \tag{12}$$

rh_{rt} is the minimum reserve of reservoir *r* at time step *t*.

- Integrality constraints:

$$z_{jkrt} \in \{0,1\} \qquad \sum_{k=1}^{\Phi_{jr}} z_{jkrt} \leq 1 \tag{13}$$

In the sequel, to alleviate notation, we write constraints C_H above in the abstract form

$C_H = C_{HH}(Q,s,V) \cap C_{HUC}(z,q,Q,s,PH)$,

where the vectors *z*, *q*, *Q*, *s*, *PH* and *V* gather the respective variables. The set C_{HH} represents constraints given by (6)-(8), modeling the reservoirs, while C_{HUC} represents the unit constraints, i.e., (9)-(13). In this abstract formulation, $\alpha = \alpha(V)$.

4.2 *Constraints involving only thermal variables* (C_T)

- Power limit for each unit:

$$pt_i^{min} u_{it} \leq pt_{it} \leq pt_i^{max} u_{it} \tag{14}$$

Here $pt_i^{min,max}$ stand for the minimum and maximum power limits of unit *i*. The binary variable u_{it} is 1 if the unit is operating at time step *t*, and it is set to 0 otherwise.

- Reserve constraints:

$$pt_i^{max} u_{it} - pt_{it} \geq rt_{it} \tag{15}$$

rt_{rt} is the reserve of unit *i* at time step *t*.

- Minimum up-time, t_i^{up}, and downtime, t_i^{down}, for each unit:

$$u_{it} = \begin{cases} 1 & \text{if} \quad 1 \leq x_{it} < t_i^{up} \\ 0 & \text{if} \quad 1 \geq x_{it} > -t_i^{down} \\ 0 \text{ or } 1 & \text{otherwise} \end{cases} \qquad x_{it} = \begin{cases} \max(x_{i,t-1}, 0) + 1, & \text{if } u_{it} = 1, \\ \min(x_{i,t-1}, 0) - 1, & \text{if } u_{it} = 0, \end{cases} \tag{16}$$

where the non zero integer variable x_{it} counts the number of time steps the unit was on previous time step *t*.

- Ramp constraints:

$$\delta_i(u_{i,t-1}, x_{it}) \leq p_{it} - p_{i,t-1} \leq \Delta_i(u_{i,t-1}, x_{it}) \tag{17}$$

$\delta_i(.)$ and $\Delta_i(.)$ are the maximum allowed variations of generation of the unit between two time steps.

In a abstract formulation, constraints in the set C_T correspond to $C_T(u,pt)$, where *u* and *pt* are vectors gathering all binary and continuous thermal variables, respectively.

4.3 *Constraints involving both hydraulic and thermal variables* (C_{HT})

- Satisfaction of demand, per time step and subsystem:

$$\sum_{i\in I_e} pt_{it} + \sum_{r\in R_e} PH_{rt} + \sum_{l\in\Omega_e} (Int_{let} - Int_{elt}) = D_{et} \tag{18}$$

The interconnected hydrothermal system is divided into subsystems, indexed by e. Accordingly, all thermal units (reservoirs) of subsystem e are gathered in the index set I_e (R_e). There are Ω_e subsystems interconnected with subsystem e, the corresponding exchange of energy, from and to subsystem l, is given by Int_{elt}. for each time step t. Finally, D_{et} is the demand of subsystem and at time t.

- Subsystems exchange limits, from e (l) to l (e), at time t, Int_{elt}^{max}, (Int_{let}^{max}):

$$0 \le Int_{let} \le Int_{let}^{max} \qquad 0 \le Int_{elt} \le Int_{elt}^{max} \tag{19}$$

In our abstract notation, the set C_{HT} is written as $C_{HT}(pt,PH,Int)$, where the vector Int gathers the subsystem exchanges.

The above description confirms the level of complexity of the optimization problem to be solved. We now address the solution strategy adopted in this work.

4. Solving the HUC Problem

The economic impact of the optimal scheduling of power plants is undeniable. Because of their solid theoretical background, LR techniques appear in this area as the preferred solution method. In particular, multipliers associated to demand constraints given by (18) are used to price energy.

The *divide to conquer* approach of LR, also called price decomposition [9], is well known. Essentially, coupling constraints are relaxed via Lagrange multipliers whose corresponding dual problem is decomposable into simpler subproblems (called local subproblems). The coordination of subproblems is then done by a master program, which finds new multipliers by making one iteration of a nonsmoth algorithm that maximizes the dual function.

There are many ways of relaxing coupling constraints. An important criterion for deciding how to proceed is the resulting duality gap, which should be the smallest possible. In this matter, the introduction of artificial variables to uncouple constraints appears as a good choice; see [14], and also [5,15] for an application to the thermal UC problem. For this reason, we apply a similar approach in this work, and derive three different decomposition schemes, adapted to different unit configurations in the Brazilian hydrothermal system.

4.1 First Decomposition Strategy – D1

In the abstract notation, the thermal HUC problem becomes:

$$\begin{aligned}&\underset{u,pt,z,q,Q,s,PH,Int}{\text{minimize}} \quad c(pt) + \alpha(V)\\ &\text{on: } C_T(pt,u) \cap C_{HT}(pt,PH,Int) \cap C_{HH}(Q,s,V) \cap C_{HUC}(z,q,Q,s,PH)\end{aligned} \tag{20}$$

To achieve decomposition, we introduce artificial variables pta and PHa, which duplicate, respectively, pt and PH. Variables pta and PHa are used in constraints C_{HT} to replace pt and PH. In addition, artificial variables Qa and sa duplicate Q and s, respectively. Qa and sa replace Q and s in C_{HH}. With these additional variables, (20) is rewritten as follows:

$$\begin{aligned}&\underset{u,pt,pta,z,q,Q,Qa,s,sa,PH,PHa,Int}{\text{minimize}} \quad c(pt) + \alpha(V)\\ &\text{on: } C_T(pt,u) \cap C_{HT}(pta,PHa,Int) \cap C_{HH}(Qa,sa,V) \cap C_{HUC}(z,q,Q,s,PH)\\ &\quad pt = pta \quad PH = PHa \quad Q = Qa \quad s = sa\end{aligned} \tag{21}$$

In (21) the newly introduced artificial constraints hold the coupling of the problem. Hence, we relax them by associating Lagrange multipliers λ_{PT}, λ_{PH}, λ_Q, λ_S and writing the corresponding dual problem[3]:

$$\begin{aligned}&\underset{\lambda_{PT},\lambda_{PH},\lambda_Q,\lambda_s}{\text{maximize}} \; \underset{u,pt,pta,z,q,Q,s,PH,PHa,Int}{\text{minimize}} \left[c(pt) + \alpha(V) + \lambda_{PT}^T(pt-pta) + \lambda_{PH}^T(PH-PHa) + \lambda_Q^T(Q-Qa) + \lambda_S^T(s-sa)\right]\\ &\qquad\qquad \text{on: } C_T(pt,u) \cap C_{HT}(pta,PHa,Int) \cap C_{HH}(Qa,sa,V) \cap C_{HUC}(z,q,Q,s,PH)\end{aligned} \tag{22}$$

Problem (22) can be rewritten as follows:

[3] From now on, the Euclidean inner product of two vectors, λ and v, will be denoted by $\lambda^T v = \Sigma_i \lambda_i v_i$.

$$\underset{\lambda=[\lambda_{PT},\lambda_{PH},\lambda_{Q},\lambda_{S}]}{\text{maximize}} \ \mathrm{D1}(\lambda) := \mathrm{D1_T}(\lambda)+\mathrm{D1_{HT}}(\lambda)+\mathrm{D1_{HH}}(\lambda)+\mathrm{D1_{HUC}}(\lambda) \tag{23}$$

where:

$$\begin{aligned} &\mathrm{D1_T}(\lambda) = \min_{u,pt} \ c(pt) + \lambda_{PT}^{\mathrm{T}} pt \\ &\qquad \text{on: } \mathrm{C_T}(pt,u) \end{aligned} \tag{24}$$

$$\begin{aligned} &\mathrm{D1_{HT}}(\lambda) = \min_{pta,PHa,Int} \ -\left[\lambda_{PT}^{\mathrm{T}} pta + \lambda_{PH}^{\mathrm{T}} PHa\right] \\ &\qquad \text{on: } \mathrm{C_{HT}}(pta,PHa,Int) \end{aligned} \tag{25}$$

$$\begin{aligned} &\mathrm{D1_{HH}}(\lambda) = \min_{Qa,sa} \ \alpha(V) - \lambda_{Q}^{\mathrm{T}} Qa - \lambda_{s}^{\mathrm{T}} sa \\ &\qquad \text{on: } \mathrm{C_{HH}}(Qa,sa,V) \end{aligned} \tag{26}$$

$$\begin{aligned} &\mathrm{D1_{HUC}}(\lambda) = \min_{z,q,Q,s,PH} \ \lambda_{PH}^{\mathrm{T}} PH + \lambda_{Q}^{\mathrm{T}} Q + \lambda_{s}^{\mathrm{T}} s \\ &\qquad \text{on: } \mathrm{C_{HUC}}(z,q,Q,s,PH) \end{aligned} \tag{27}$$

In the LR approach, the primal problem (20) is replaced by the dual problem (23) whose objective function D1(λ), can be split as the sum of four terms, corresponding to subproblems (24)-(27). Subproblem (24) is a nonlinear optimization problem with continuous and binary variables, coupled along time steps, but not along plants. It can be solved by a classic Dynamic Programming method; as in [1],[5]. Subproblem (25) is a standard linear programming problem, coupled along plants, but not along times steps, which can be solved by any Linear Programming (LP) commercial solver. Subproblem (26) is also an LP problem, coupled both in time and space via the stream-flow constraints given by (6). Even though (26) can be large-scale, it can still be solver by an LP solver. Finally, subproblem (27) is a nonlinear mixed-integer optimization problem, uncoupled both in time and units. This subproblem corresponds to the commitment of hydro units, for a given reservoir and time step. The higher $J(r)$ and Φ_{jr} (the number of units in the reservoir and of operating zones, respectively) are, the bigger computational effort will be required to solve (27).

Each sub-subproblem in (27), for each time step and for a given power plant, is a mixed-integer NLP problem, with binary variables corresponding to different operating modes in the plant. The total number of possible operating modes is given by the product of all combinations of the operating modes of all the units composing the plant. Each combination of a unit is a configuration where the corresponding binary variables are fixed to one of the feasible values. Once the binary values are fixed, the problem becomes a nonlinear program, whose size is dependent on $J(r)$.

Generally, hydropower plants have identical units, and each unit has a single operating zone. In this case, the total number of modes is no longer $2^{J(r)}$, but $J(r)+1$ and, thus, a complete enumeration of modes seems a good strategy. Sometimes, however, there are power plants with many different types of units, and several operating modes. For these configurations, an enumeration procedure may become too expensive from the computational point of view. We now introduce an alternative decomposition scheme, adapted to such situations.

4.2 *Second Decomposition Strategy* – D2

In order to avoid the enumerative process required to solve subproblem (27), we eliminate the coupling between the binary variable z and the continuous variables $[q,Q,s]$ which appear in $\mathrm{C_{HUC}}$. Therefore, we rewrite (20) as follows:

$$\begin{aligned} &\underset{u,pt,z,q,Q,s,PH,Int}{\text{minimize}} \ c(pt) + \alpha(V) \\ &\text{on: } \mathrm{C_T}(pt,u) \cap \mathrm{C_{HT}}(pt,PH,Int) \cap \mathrm{C_{HH}}(Q,s,V) \cap \mathrm{C_{HUCa}}(q,Q,s,PH) \cap \mathrm{C_{HUCb}}(z,q,Q,s) \cap \mathrm{C_{HUCres}}(z,PH) \end{aligned} \tag{28}$$

Now the set $\mathrm{C_{HUCa}}$ gathers constraints given by (9) and (11), $\mathrm{C_{HUCb}}$ contains (10) and (13) and $\mathrm{C_{HUCres}}$ corresponds to the reserve constraint (12). Besides the artificial variables used in (21), we use pha to replace the hydro production function $ph(q,Q,s)$ in the set $\mathrm{C_{HUCb}}$, and rewrite (28) as:

$$\begin{aligned} &\underset{u,pt,pta,z,q,Q,Qa,s,sa,PH,PHa,Int}{\text{minimize}} \ c(pt) + \alpha(V) \\ &\text{on: } \mathrm{C_T}(pt,u) \cap \mathrm{C_{HT}}(pta,PHa,Int) \cap \mathrm{C_{HH}}(Qa,sa,V) \cap \mathrm{C_{HUCa}}(q,Q,s,PH) \cap \mathrm{C_{HUCb}}(z,pha) \cap \mathrm{C_{HUCres}}(z,PH) \\ &\quad pt = pta \quad PH = PHa \quad Q = Qa \quad s = sa \quad ph(q,Q,s) = pha \end{aligned} \tag{29}$$

Now not only the artificial constraints, but also the constraint set C_{HUCres} keep the problem coupled. Hence, we relax these constraints by introducing multipliers λ_{PT}, λ_{PH}, λ_{Q}, λ_{S}, λ_{ph} and λ_{Res} and writing the dual problem:

$$\begin{aligned}\underset{\lambda_{PT},\lambda_{PH},\lambda_Q,\lambda_s}{\text{maximize}}\ \underset{u,pt,pta,z,q,Q,Qa,s,sa,PH,PHa,Int}{\text{minimize}}\ & \Big[c(pt)+\alpha(V)+\lambda_{PT}^{\mathrm{T}}(pt-pta)+\lambda_{PH}^{\mathrm{T}}(PH-PHa)+\lambda_{Q}^{\mathrm{T}}(Q-Qa)+\lambda_{S}^{\mathrm{T}}(s-sa)+\\ & \lambda_{ph}^{\mathrm{T}}[ph(q,Q,s)-pha]+\lambda_{\mathrm{Res}}^{\mathrm{T}}[PH-ph^{\max}z+rh]\Big]\\ & \text{on: } \mathrm{C_T}(pt,u)\cap\mathrm{C_{HT}}(pta,PHa,Int)\cap\mathrm{C_{HH}}(Qa,sa,V)\cap\mathrm{C_{HUCa}}(q,Q,s,PH)\cap\mathrm{C_{HUCb}}(z,pha)\end{aligned} \tag{30}$$

Here, $ph^{\max}$ and rh are vectors corresponding to a unit's maximum capacity and to reserve levels for the plant, respectively. Separability in (30) has now the expression:

$$\underset{\lambda=[\lambda_{PT},\lambda_{PH},\lambda_Q,\lambda_S,\lambda_{ph},\lambda_{\mathrm{Res}}]}{\text{maximize}}\ \mathrm{D2}(\lambda):=\mathrm{D1_T}(\lambda)+\mathrm{D1_{HT}}(\lambda)+\mathrm{D1_{HH}}(\lambda)+\mathrm{D2_{HUCa}}(\lambda)+\mathrm{D2_{HUCb}}(\lambda)+\lambda_{\mathrm{Res}}^{\mathrm{T}}rh \tag{31}$$

where $\mathrm{D1_T}(\lambda)$, $\mathrm{D1_{HT}}(\lambda)$, $\mathrm{D1_{HH}}(\lambda)$ are the dual functions from (24), (25) and (26). The remaining terms are:

$$\begin{aligned}\mathrm{D2_{HUCa}}(\lambda)=\min_{q,Q,s,PH}\ & \lambda_{\mathrm{Res}}^{\mathrm{T}}PH+\lambda_{ph}^{\mathrm{T}}ph(q,Q,s)+\lambda_{Q}^{\mathrm{T}}Q+\lambda_{s}^{\mathrm{T}}s\\ & \text{on: } \mathrm{C_{HUCa}}(q,Q,s,PH)\end{aligned} \tag{32}$$

$$\begin{aligned}\mathrm{D2_{HUCb}}(\lambda)=\min_{z,pha}\ & -\left[\lambda_{ph}^{\mathrm{T}}pha+\lambda_{\mathrm{Res}}^{\mathrm{T}}ph^{\max}z\right]\\ & \text{on: } \mathrm{C_{HUCb}}(z,pha)\end{aligned} \tag{33}$$

Subproblem (32) is an NLP problem (with only continuous variables), for each time step and power plants. Subproblem (33) is a mixed-integer linear program on variables corresponding to a single unit, which can be solved by enumeration of the operating zones.

4.3 *Third Decomposition Strategy* – D3

Our last decomposition combines D1 and D2. Accordingly, we employ D1 for those plants with a reduced number of units and operating zones, while D2 is applied for plants for which D1 would be too expensive. We split the hydro plants index set into R(Γ) and R(Ξ), corresponding, respectively, to plants where (27) and (32)-(33) is applied:

$$\begin{aligned}&\underset{u,pt,z,q,Q,s,PH,Int}{\text{minimize}}\ c(pt)+\alpha(V)\\ &\text{on: } \mathrm{C_T}(pt,u)\cap\mathrm{C_{HT}}(pt,PH,Int)\cap\mathrm{C_{HH}}(Q,s,V)\cap\mathrm{C_{HUC}^{r\in R(\Gamma)}}(z,q,Q,s,PH)\cap\mathrm{C_{HUCa}^{r\in R(\Xi)}}(q,Q,s,PH)\cap\\ &\quad \mathrm{C_{HUCb}^{r\in R(\Xi)}}(z,q,Q,s)\cap\mathrm{C_{HUCres}^{r\in R(\Xi)}}(z,PH)\end{aligned} \tag{34}$$

We proceed like for D1 and D2, but splitting the reservoir index sets:

$$\begin{aligned}&\underset{u,pt,pta,z,q,Q,Qa,s,sa,PH,PHa,Int}{\text{minimize}}\ c(pt)+\alpha(V)\\ &\text{on: } \mathrm{C_T}(pt,u)\cap\mathrm{C_{HT}}(pta,PHa,Int)\cap\mathrm{C_{HH}}(Qa,sa,V)\cap\mathrm{C_{HUC}^{r\in R(\Gamma)}}(z,q,Q,s,PH)\cap\mathrm{C_{HUCa}^{r\in R(\Xi)}}(q,Q,s,PH)\cap\\ &\quad \mathrm{C_{HUCb}^{r\in R(\Xi)}}(z,q,Q,s)\cap\mathrm{C_{HUCres}^{r\in R(\Xi)}}(z,PH)\\ &\quad pt=pta\quad PH=PHa\quad Q=Qa\quad s=sa\quad ph(q,Q,s)=pha,r\in\Gamma(\Xi)\end{aligned} \tag{35}$$

After relaxation, the dual problem of (35) is:

$$\underset{\lambda=[\lambda_{PT},\lambda_{PH},\lambda_Q,\lambda_S,\lambda_{ph},\lambda_{\mathrm{Res}}]}{\text{maximize}}\ \mathrm{D3}(\lambda):=\mathrm{D1_T}(\lambda)+\mathrm{D1_{HT}}(\lambda)+\mathrm{D1_{HH}}(\lambda)+\mathrm{D1_{HUC}}(\lambda)+\mathrm{D2_{HUCa}}(\lambda)+\mathrm{D2_{HUCb}}(\lambda) \tag{36}$$

The dual functions $\mathrm{D1_T}(\lambda)$, $\mathrm{D1_{HT}}(\lambda)$, $\mathrm{D1_{HH}}(\lambda)$ are those in (24), (25) and (26), respectively. The dual function $\mathrm{D1_{HUC}}(\lambda)$ from (27) only applies for reservoirs with index $r \in R(\Gamma)$. The remaining dual functions in (36) apply to reservoirs $r \in R(\Xi)$, and are given by subproblems (32)-(33).

4.4 *Nonlinear Programming Subproblems*

A crucial issue for an effective application of LR is the fast resolution of subproblems giving the dual function for a fixed multiplier λ. Since many of such subproblems are nonlinear programs, we implemented a Sequential Quadratic Programming (SQP) quasi-Newton method. More precisely, each iteration k of the algorithm generates a direction p^k by solving the quadratic programming problem:

$$\begin{aligned}&\underset{p^k}{\text{minimize}}\quad \nabla f(x^k)^{\mathrm{T}}p^k+0.5\left(p^k\right)^{\mathrm{T}}M^kp^k\\ &\text{s.t.:}\qquad \nabla c_e(x^k)^{\mathrm{T}}p^k+c_e(x^k)=0\qquad \nabla c_i(x^k)^{\mathrm{T}}p^k+c_i(x^k)\le 0\end{aligned} \tag{37}$$

Here, $f(x^k)$, $c_e(x^k)$ and $c_i(x^k)$ represent, respectively, the objective and equality and inequality constraint functions at a point x^k. The matrix M^k estimates the Lagrangian Hessian for the NLP, L^k. In order to avoid the calculation of second-order derivatives, and to preserve positive definiteness of the sequence of quasi-Newton matrices, we use a BFGS (Broyden-Fletcher-Goldfarb-Shanno) [16,17], formula, appended with a Powell correction [16]:

$$M^{k+1} = M^k - \frac{M^k s^k (s^k)^{\mathrm{T}} M^k}{(s^k)^{\mathrm{T}} M^k s^k} + \frac{r^k (z^k)^{\mathrm{T}}}{(s^k)^{\mathrm{T}} r^k} \tag{38}$$

where:

$$s^k = x^{k+1} - x^k \qquad\qquad z^k = \nabla_x L^{k+1} - \nabla_x L^k \tag{39}$$

$$r^k = \theta^k z^k + (1-\theta^k) M^k s^k \tag{40}$$

$$\theta^k = \begin{cases} 1 & se \quad (s^k)^{\mathrm{T}} z^k \geq 0{,}20 (s^k)^{\mathrm{T}} M^k s^k \\ \dfrac{0{,}8 (s^k)^{\mathrm{T}} M^k s^k}{(s^k)^{\mathrm{T}} M^k s^k - (s^k)^{\mathrm{T}} z^k} & se \quad (s^k)^{\mathrm{T}} z^k < 0{,}20 (s^k)^{\mathrm{T}} M^k s^k \end{cases} \tag{41}$$

Globalization of the method is achieved by performing a line search on the following function[4]:

$$\phi(x^k, \sigma) = f(x^k) + \sigma^k \left\| c(x^k)^\# \right\|_\infty \tag{42}$$

known as Han's merit function [9]. Since $\phi(x^k,\sigma^k)$ is an exact penalty function, there is a finite positive value $\underline{\sigma}^k$ such that an unconstrained minimum of $\phi(x^k,\sigma^k)$ solves the original NLP for all $\sigma^k \geq \underline{\sigma}^k$. We update the parameter σ^k accordingly. More precisely, the directional derivative of the merit function along the direction satisfies the relation:

$$D[\phi(x^k, \sigma^k); p^k] \leq \nabla f(x^k)^{\mathrm{T}} p^k - \sigma^k \left\| c(x^k)^\# \right\|_\infty \leq (p^k)^{\mathrm{T}} M^k p^k + (\zeta^k)^{\mathrm{T}} c(x^k) - \sigma^k \left\| c(x^k)^\# \right\|_\infty \tag{43}$$

where ζ is the Lagrange multiplier associated with constraints in (37). For any stationary point of (42), such as p^k, it can be shown that the estimate in (43) gives a descent direction for ϕ if M^k is positive definite and σ^k is updated in order to satisfy:

$$\sigma^k \geq \left\| \zeta^k \right\|_\infty + \bar{\delta} \qquad\qquad \bar{\delta} > 0 \tag{44}$$

We exit the line search when the Armijo [18] condition is satisfied:

$$\phi^k (x^k + \alpha p^k) \leq \phi^k (x^k) + \omega \alpha \Delta^k \tag{45}$$

here, $\omega \in]0,1/2[$ and α is the positive stepsize. Ideally, Δ^k should be the exact value of the directional derivative. We estimate it by the upper bound from (43):

$$\Delta^k := \nabla f(x^k)^{\mathrm{T}} p^k - \sigma^k \left\| c(x^k)^\# \right\|_\infty \tag{46}$$

Finally, we add an extra term to p^k in order to avoid Maratos effect [19]. This phenomenon may impair the superlinear local convergence rate by rejecting unit stepsizes when close to a solution. The corrected direction $p^{\underline{k}}$, as shown in [9], ensure that asymptotically there is enough constraint reduction. The extra constraint evaluations required by our correction is compensated by the robustness and efficiency gained by the method.

5. Numerical Results

We assess the three decomposition schemes on a real-life hydroelectric configuration extracted from the Southern region of the Brazilian hydrothermal power system. More precisely, we consider a system with 121 generating units whose maximum installed capacity is 31.129,2 MW[5]. Figure 2 reports the data for the system, where power plants numbered #3, #4, #6 and #14 have production functions independent of spillage. Values between brackets in Fig. 2 correspond to water travel times, expressed in hours. For this configuration, the biggest power plants are #7 and #16, with 20 units each one, while the smallest plant, #14, has only 2 units. The planning horizon of two days is discretized in hourly time steps, yielding T=48. Initial reservoir

[4] The symbol # is used to denote only active constraints at x^k.

[5] This amount corresponds to about 49% of the total hydraulic capacity of Brazil.

volumes were taken at 50% of the usable volumes, while the inflows were considered null. We do not address here the dual solution in detail, we refer to [5] for this subject. Instead, we fixed Lagrange multipliers for each reservoir and time step, λ_{PHrt}, and use a proximal quasi-Newton variant of a bundle method to optimize the remaining multipliers; see Ch. 9 in [9]. The values for λ_{PHrt} are chosen based on generation costs associated with typical demand curves, i.e., with higher values for peak times with high demand[6].

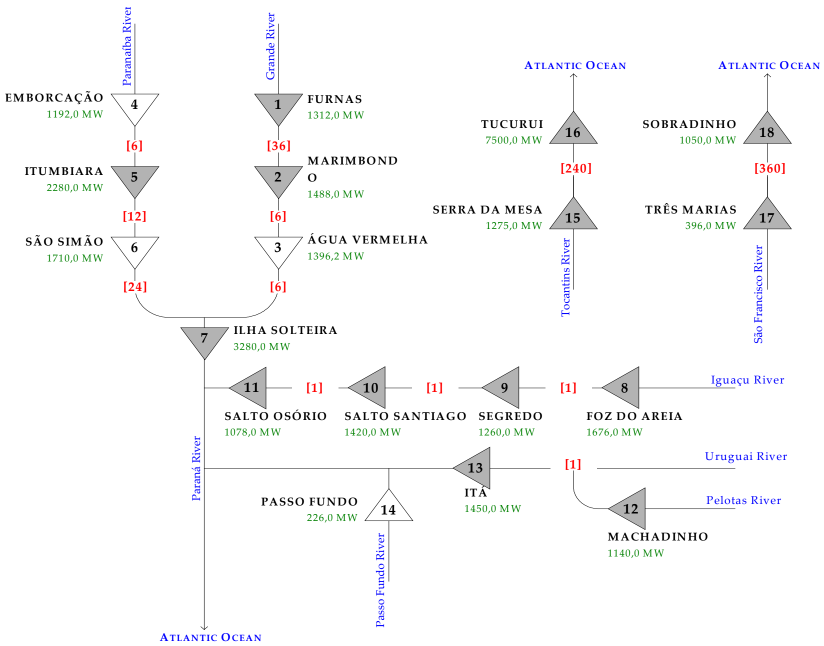

Figure 2 – Hydroelectric Configuration.

We implemented the three dual subproblems $D1_{UCH}$, $D2_{UCH}$, $D3_{UCH}$, corresponding respectively to (27), (32)-(33) and (36). For the dual solution we use N1CV2 code; see [20]. Subproblem $D1_{UCH}$ has $(R+Rv)\times T$ variables, where Rv denotes the number of reservoirs with production function depending on spillage. Since T=48, R=18 and Rv =14, subproblem (27) has 1536 variables. In (32)-(33) subproblem $D2_{UCH}$ has $(2\times R+Rv+ngmix)\times T$=8208 variables, where $ngmix$ = 121 denotes the total number of units in the mix. The size of subproblem $D3_{UCH}$ depends on the decomposition scheme. The LP given by $D1_{HH}$ (26), as well as other LPs are solved using ILOG CPLEX 7.1 solver. For $D1_{UCH}$ there are, in addition, 6288 NLP problems. Finally, for $D2_{UCH}$ there are $R\times T$ = 864 NLP and $T\times ngmix$ = 5808 easy mixed-integer LP problems[7].

For $D1_{UCH}$ we found an optimal value of $ -17.339.230,0, after 175 iterations, that took 180 minutes of CPU times in a Pentium III 550 MHz computer with 128 Mb of RAM memory. For $D2_{UCH}$ the optimal value found was $ -17.450.168,0, after 325 iterations in 50 minutes. Since the dual value in $D2_{UCH}$ is smaller than the one from $D1_{UCH}$, and dual values give lower bounds for the primal optimal value, we can conclude that primal variables associated with $D1_{UCH}$ are better than those associated with $D2_{UCH}$.

To further assess the previous remark, we now consider in more detail some selected primal variables, for some specific reservoirs. Figures 3 and 4 show the values for Q and Qa obtained at the last iteration of both $D1_{UCH}$ and $D2_{UCH}$ for Sobradinho #18 power plant. We also show the value of λ_{PHrt} (price) used to solve the subproblems.

---

[6] These costs presented values between 0 and 45 $/MW.

[7] These are indeed easy to solve problems, with only two variables (one integer, one continuous), and two constraints.

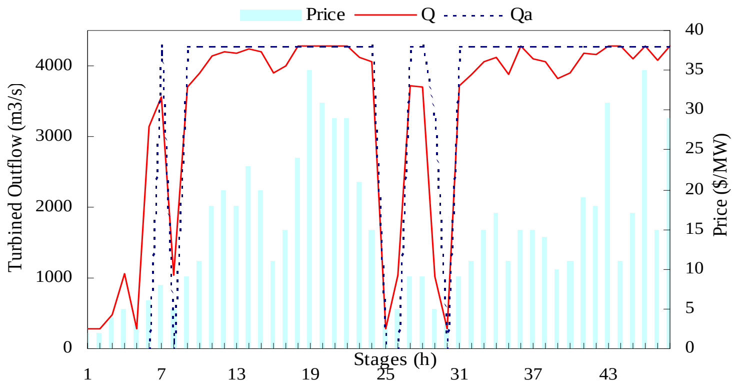

Figure 3 – Sobradinho Power Plant #18 – D1$_{UCH}$.

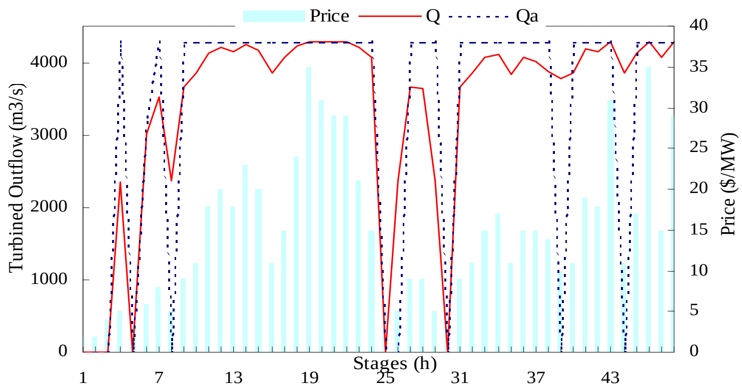

Figure 4 – Sobradinho Power Plant #18 – D2$_{UCH}$.

It can be seen in the figures that Qa takes mostly two values: 0 and 4278 m^3/s (maximum value). This *bang-bang* behaviour is explained by the linear nature of subproblem (26). The values for Q exhibit a different behaviour, closer to the price profile, i.e., to λ_{PHrt}. From the comparison of D1$_{UCH}$ and D2$_{UCH}$ we see that the relaxed primal constraint[8] is more violated for D2$_{UCH}$, a problem that contains a higher number of relaxed constraints. However, infeasibility becomes smaller for time steps with bigger λ_{PHrt}, i.e., for peaks of demand. For these time steps, primal points obtained with D2$_{UCH}$ are good approximations to those from D1$_{UCH}$, requiring a computational effort that can be up to 3 times bigger. For lower prices, both problems give primal points that are even more infeasible, the worse values being associated with D2$_{UCH}$.

The results observed for Sobradinho power plant are typical for all the plants in the mix, with variations in the computed primal infeasibility. In general, we observed that for reservoirs downstream, that tend to operate with outflows near to the nominal values, primal points obtained from D2$_{UCH}$ were close to those from D1$_{UCH}$. We report this behaviour in Figures 5 and 6, with the results for Ilha Solteira – 7 power plant.

[8] The difference between Q and Qa.

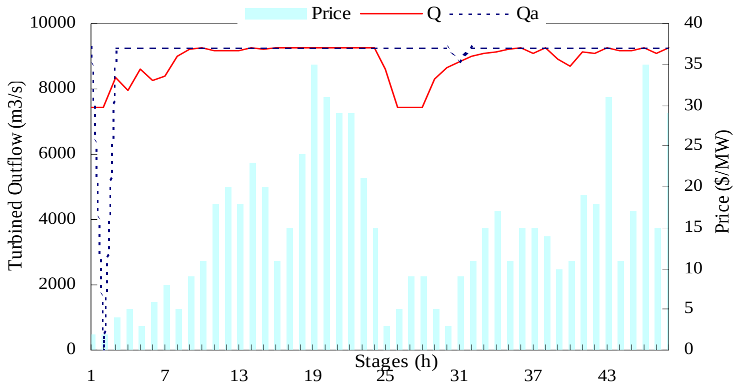

Figure 5 – Ilha Solteira Power Plant #7 – D1$_{UCH}$.

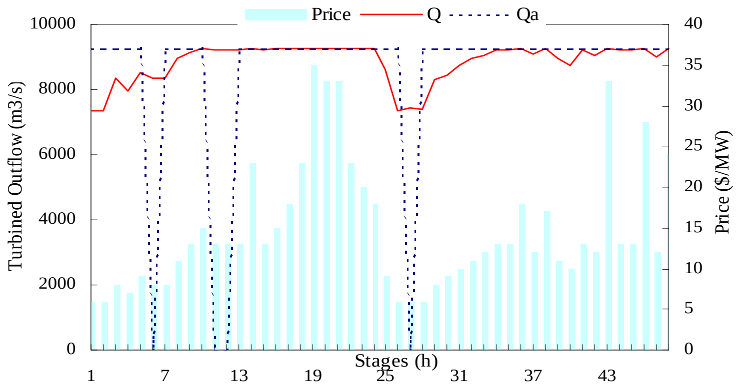

Figure 6 – Ilha Solteira Power Plant #7 – D2$_{UCH}$.

We conclude from our experiments that, in terms of primal solutions, subproblem D1$_{UCH}$ is a better option. However, since average hourly CPU times for D1$_{UCH}$ are 3.5 higher than D2$_{UCH}$, the enumerative process required by D1$_{UCH}$ should not be used for power plants with complex configurations, such as #7 and #16. In Table 1 we report the main results for the third decomposition scheme, D3$_{UCH}$, where for R(Ξ)={7,16} we applied the scheme corresponding to (32)-(33). We also give, for comparison purposes, the previously obtained values for D1$_{UCH}$ and D2$_{UCH}$.

Table 1 – Numerical Results.

Dual Problem	D1$_{UCH}$	D2$_{UCH}$	D3$_{UCH}$
Cost ($)	-17.339.230	-17.450.168	-17.359.077
Iteration	175	275	220
Time (minutes)	180	50	70
Variables	1536	8208	2592
Number of PNL[9]	6288	864	4560

The computational effort required for solving subproblems involving the big plants #7 and #16 (with identical 20 units each), is clear in Table 1. Even though D3$_{UCH}$ [10] needs to solve 27,50% NLP problems less than D1$_{UCH}$, in terms of CPU times the gain was of 61,11%. Another important matter shown by Table 1 concerns dual optimal values: the (absolute value) difference between D2$_{UCH}$ and D1$_{UCH}$ is of $ 110.938, while the difference between D3$_{UCH}$ and D1$_{UCH}$ is $ 19.847, i.e., about 5.5 smaller.

Finally, it should be kept in mind that the approach presented here only addresses the dual solution, whose associated *optimal* primal points are infeasible for the original problem. In order to recover primal feasibility, a purification-like phase should be executed afterwards. Such processes are often based on heuristics depending on the particular problem structure; see for example [21,22]. In particular, for the Brazilian case,

[9] Total number of NLP problems solved at each iteration.
[10] Ilha Solteira #7 and Tucuruí #16 power plants are the only ones where (27), requiring the enumerative process, was not employed; we use (32)-(33) instead.

general-purpose combinatorial optimization heuristics are not suitable. An augmented Lagrangian technique seems in this case better, we refer to [5] for a description of this technique in a similar context.

5.1 Sequential Quadratic Programming Algorithm

In our work, NLP problems have ng+2 variables and 2ng+2 constraints, where ng is the number of generating units in the considered configuration (in our example $ng \leq 20$). SQP algorithms do not need starting feasible points; in our implementation starting points only satisfy constraints (6) and (10) (i.e., both the penstock stream-flow equations and power limit constraints), but not the reserve constraint (12). In addition, to prevent the starting quadratic program (37) to be infeasible, we introduced additional constraints on the direction obtained from (37), aimed at satisfying physical operating bounds. Such bounds are related to each unit maximum turbined outflow, as well as maximum spillage levels. We mention that more sophisticated alternatives would be possible, for instance considering active constraints as [23], or introducing slack variables in a trust-region SQP, as in [24].

To solve each quadratic program (37), we use the Fortran code PLCBAS, an active set QP solver described in [25]. Once the direction is computed, the algorithm checks validity of (45) and then performs, if needed, two different correction strategies to avoid Maratos effect. More precisely, we first implemented a correction on the constraint reduction. With this correction, the algorithm was sometimes stalling and stopped after having attained the maximum number of simulations. We then implemented a second correction, to be used only after a certain number of simulations has been done inside of the same iteration. This correction accepts higher values of the merit function, if close to a solution. With this combination of corrections, the efficiency of the method was sensibly improved.

We use the following stopping test:

$$opt\_test = \left\|\nabla L^k\right\|_\infty + \left\|c(x^k)^\#\right\|_\infty \leq eps \tag{47}$$

with eps = 1,0×10^{-8}. Emergency exits, after 300 iterations of 600 simulations were also implemented. The observed average values for achieving convergence were of 8 iterations and 10 simulations.

In order to assess our self-made algorithm, we compare its performances the NLP solver Easy!, available for academic applications from the Optimization Group of Campinas [26]. This solver uses the augmented Lagrangian method described in [27]. We obtained identical results, with inferior CPU times than those employed by Easy!. However, we did not take advantage of all resources available in Easy! solver[11]. Furthermore, our own implementation was tailored for the structure of our specific problem. We mention that more sophisticated alternatives would be possible, for instance considering active constraints as [23], or introducing slack variables in a trust-region SQP, as in [24].

6 Concluding Remarks

We address in this work the problem of optimal commitment of hydraulic generating units in a hydrothermal power system. Two main topics were discussed, namely the modeling and solution strategy for the hydraulic problem. We gave a detailed modeling for the production function of each hydro unit, which takes into account the effect of variable efficiency rates, hydraulic losses, tailrace levels, as well as multiple operating zones. With respect to the solution methodology, we gave an LR decomposition approach using variable duplication to uncouple difficult constraints. We assessed the decomposition method by implementing our approach and testing it on a real-life hydraulic configuration, extracted from the Brazilian system. Our implementation focused on the hydraulic subproblems. We analyzed in detail the practical applicability of three decomposition schemes, in terms of CPU times and obtained primal points. For our configuration, the enumerative process appeared preferable for most power plants. For the whole Brazilian system, double in size to our test, this approach might become too expensive and a combined strategy, like the one given in our third decomposition scheme, may be preferable.

[11] Easy! solver is more efficient when analytic derivatives are available for the simulations; otherwise a finite difference approximation needs to be estimated at each simulation.

7. Bibliography

[1] Ruzic, S., Rajakovic, N., Vuckovic, A.; A Flexible Approach to Short-Term Hydro-Thermal Coordination, PART I: Problem Formulation and General Solution Procedure, *IEEE Transactions on Power Systems*, Volume 11, Number 3, August, 1996.

[2] C. Li, E. Hsu, A. J. Svoboda, C. L. Tseng, R. B. Johnson; Hydro Unit Commitment in Hydro-Thermal Optimization, *IEEE Transactions on Power Systems*, Volume 12, Number 2, May, 1997.

[3] X. Guan, A. Svoboda, C. A. Li; Scheduling Hydro Power Systems with Restricted Operating Zones and Discharges Ramping Constraints, *IEEE Transactions on Power Systems*, Volume 14, Number 1, February 1999.

[4] E. Ni, X. Guan, R. Li; Scheduling Hydrothermal Power Systems with Cascaded and Head-Dependent Reservoirs, *IEEE Transactions on Power Systems*, Volume 14, Number 3, August, 1999.

[5] A. Belloni, A. L. D. S. Lima, M. E. P. Maceira, C. A. Sagastizábal; Bundle Relaxation and Primal Recovery in Unit Commitment Problems – The Brazilian Case, *Annals of Operations Research Special - Volume on OR models for Energy Policy, Planning and Management*, Kluwer Academic Publishers, Netherlands, 2001.

[6] G. B. Sheble and G. N. Fahd; Unit Commitment Literature Synopsis, *IEEE Transactions on Power Systems*, Volume 9, Number 1, February, 1994.

[7] P. T. Boggs and J. W. Tolle; Sequential Quadratic Programming, *Acta Numerica*, pp. 1- 51, 1996.

[8] J. Nocedal, S. J. Wright; *Numerical Optimization*, Springer Series in Operations Research, 1999.

[9] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, C. Sagastizábal; *Numerical Optimization. Theoretical and Practical Aspects*, Universitext, Springer-Verlag, Berlin, 2002.

[10] D. P. Bertsekas; *Nonlinear Programming*, Athena Scientific, 2nd Edition, Belmont, MA, 1999.

[11] A. Arce, T. Ohishi, S. Soares; Optimal Dispatch of Generating Units of the Itaipú Hydroelectric Plant, *IEEE Trans. Power Systems*, v. 17, Number 1, February, 2002.

[12] F. J. Heredia, N. Nabona; Optimum Short-Term Hydrothermal Scheduling with Spinning Reserve Through Network Flows, *IEEE Trans. Power Systems*, v. 10, Number 3, August, 1995.

[13] E. L. da Silva, E. C Finardi; Parallel Processing Applied To the Planning of Hydrothermal Systems. *IEEE Transactions on Parallel and Distributed Systems*, v. 14, pp. 721-729, 2003.

[14] Lemaréchal, C., Renaud, A.; A Geometric Study of Duality Gaps, With Applications, *Mathematical Programming*, 90 (3), pp. 399-427, 2001.

[15] Lemaréchal, C., Pellegrino, F., Renaud, A., Sagastizábal, C.; Bundle Methods Applied to the Unit Commitment Problem, *System Modeling and Optimization*, pp. 395-402, 1996.

[16] Han, S. P.; Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems, *Mathematical Programming*, 11:263-282, 1976.

[17] Powell, M. J. D.; A Fast Algorithm for Nonlinearly Constrained Optimization Calculations, *Numerical Analysis*, pp. 144-157, Springer-Verlag, Berlin, 1978.

[18] Armijo, L.; Minimization of Functions Having Lipschitz Continuous First Partial Derivates, *Pacific Journal of Mathematics*, 16:1-3, 1966.

[19] Maratos, N.; Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization, Ph. D. Thesis, University of London, 1978.

[20] Lemaréchal, C., Sagastizábal, C.; Use of the Code N1CV2, Inria, France, 2002.

[21] Blondel, H.; Renaud, A.; Décomposition Par Les Prix et Lagrangien Augmenté: Une Association Efficace Pour le Traitement des Problémes de Production, Bulletin EDF, Série C, 4, pp. 23-43, 1991.

[22] Zhuang, F., Galiana, F. D.; Towards a More Rigorous and Practical Unit Commitment by Lagrangian Relaxation, *IEEE Transactions on Power Systems*, Volume 3, Number 2, May, 1988.

[23] Lawrence, C. T.; A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm, Ph.D. Thesis, University of Maryland, USA, 1998.

[24] Celis, M. R., Dennis, J. E., Tapia, R.A., A Trust-Region Strategy for Nonlinear Equality Constrained Optimization, *Numerical Optimization,* SIAM, Philadelphia, pp.71-82, 1985.

[25] Casas, E., Pola, C.; A Sequential Generalized Quadratic Programming Algorithm Using Exact l_1 Penalty Functions, *Optimization Methods and Software*, 2:145-176, 1993.

[26] Internet address: http://www.ime.unicamp.br/~martinez/software.htm

[27] Friedlander, A., Martinez, J. M., Santos, S. A.; A New Trust Region Algorithm for Bound Constrained Minimization, *Applied Mathematics and Optimization* 30, pp. 235-266, 1994.