Long-term electric power planning in liberalized markets using the Bloom and Gallant formulation

Narcís Nabona, Adela Pagès

Dept. Statistics and Operations Research.
Univ. Politècnica de Catalunya, 08028 Barcelona, Spain.
e-mails: {narcis.nabona, adela.pages}@upc.es.

1 Introduction and Motivation

Long-term generation planning is a key issue in the operation of an electricity generation company. Its results are used both for budgeting and planning fuel acquisitions and to provide a framework for short-term generation planning.

The long-term problem is a well-known stochastic optimisation problem because several of its parameters are only known as probability distributions (for example: load, availability of thermal units, hydrogeneration and generations from renewable sources in general).

A long-term planning *period* (e.g., a natural year) is normally subdivided into shorter *intervals* (e.g., a week or a month), for which parameters (e.g., the load-duration curve) are known or predicted, and optimized variables (e.g., the expected energy productions of each generating unit) must be found.

Predicted load-duration curves (LDC's) — equivalent to cumulative probability load distributions — for each interval are used as data for the problem, which is appropriate since load uncertainty can be suitably described through the LDC. The probability of failure for each thermal unit is assumed to be known.

Bloom and Gallant [2] proposed a linear model (with an exponential number of inequality constraints) and used an *active set* methodology [9] to find the optimal way of matching the LDC of a single interval with thermal units only, when there are load-matching and other operational non-load-matching constraints. These could be, for example, limits on the availability of certain fuels, or environmental maximum emission limits. The optimal *loading order* obtained with Bloom and Gallant's method may include permutations with respect to the *merit order* and *splittings* in the loading of units [2, 8]. In this way the energies generated satisfy the limitations imposed by the non-load-matching constraints while having the best possible placement, with respect to generation cost, in the matching of the LDC.

When the long-term planning power problem is to be solved for a generation company operating in a competitive market, the company has not a load of its own to satisfy, but it bids the energies of its units to a *market operator*, who selects the lowest-price ones among biding companies to match the load. In this case, the scope of the problem is no longer that of the generation units of a single generation company but that of all units of all companies biding in the same competitive market, matching the load of the whole system. This makes planning problems much larger than before and is a reason for developing more efficient codes to solve them.

The Bloom and Gallant model has been successfully extended to multi-interval long-term planning problems using either the active-set method [9], the Dantzig-Wolfe column-generation method [4, 14], or the Ford-Fulkerson column-generation (FFcg) method [5, 12]. The FFcg and the Dantzig-Wolfe procedures have many common steps. The model has also been coded using the modeling language *AMPL* [6] and has been solved with a linear/quadratic programming package *Cplex 7.5* [3] as carried out in [11] for a single interval.

A quadratic model is put forward here to formulate the long term profit maximization of generation companies in a liberalized electricity market [10] and the performance of several solutions procedures for solving this problem is compared [13].

2 The load-duration curve

The LDC is the most sensible way to represent the load of a future interval. The main features of an LDC (corresponding to the $i^{\underline{\text{th}}}$ interval) can be described through 5 characteristics: the duration T^i, the peak load power $\widehat{P}^i$, the base load power $\underline{P}^i$, the total energy $\widehat{E}^i$ and the shape, which is not a single parameter and is usually described through a table of durations and powers, or through a function.

The LDC for future intervals must be predicted. For a past interval, for which the hourly load record is available, the LDC is equivalent to the load over time curve sorted in order of decreasing power. It should be noted that in a *predicted* LDC, random events such as weather, shifts in consumption timing, etc., that cause modifications of different signs in the load tend to cancel out, and that the LDC keeps all the power variability of the load.

3 Thermal Units

As far as loading an LDC is concerned, the relevant parameters of a thermal unit are:

- ⋆ *power capacity*: (C_j for the $j^{\underline{\text{th}}}$ unit) maximum power output (MW) that the unit can generate
- ⋆ *outage probability*: (q_j for the $j^{\underline{\text{th}}}$ unit) probability of a unit not being available when it is required to generate
- ⋆ *linear generation cost*: ($\widetilde{f}_j$ for the $j^{\underline{\text{th}}}$ unit) production cost in € /MWh

Other associated concepts are:

- ⋆ *merit order*: units are ordered according to their efficiency in generating electric power (€ /MWh); all units will work at their maximum capacity since no unit should start to generate until the previous unit in the merit order is generating at its maximum capacity,
- ⋆ *loading order*: units will have load allocated to them in a given order; loading order and merit order may not coincide when there are other constraints to be satisfied.

4 Matching the load-duration curve

Due to the outages of thermal units (whose probability is >0), the LDC does not coincide with the estimated production of thermal units. It is usual for the installed capacity to be higher than the peak load: $\sum_{j=1}^{n_u} C_j > \widehat{P}$.

The *generation-duration curve* is the expected production of the thermal units over the time interval to which the LDC refers. The energy generated by each unit is the slice of area under the generation-duration curve which corresponds to the capacity of the thermal unit.

The probability that there are time lapses within the time interval under consideration, where, due to outages, there is not enough generation capacity to cover the current load, is not null. Therefore, *external energy* (from other interconnected utilities) will have to be imported and paid for at a higher price than the most expensive unit in ownership. The peak power of the generation-duration curve is $\sum_{j=1}^{n_u} C_j + \widehat{P}$ and the area above power $\sum_{j=1}^{n_u} C_j$ is the external energy.

4.1 Convolution method of finding the generation-duration curve

The loading of thermal units in an LDC was first formulated in [1] and practical procedures to compute the expected generation can be found in [15]. Analytically, given the probability density function of load $p(x)$, the cumulative load distribution function $L_0(x)$ (see Fig. 8.1) is calculated as follows:

$$L_0(x) = 1 - \int_0^x p(y)\,dy$$

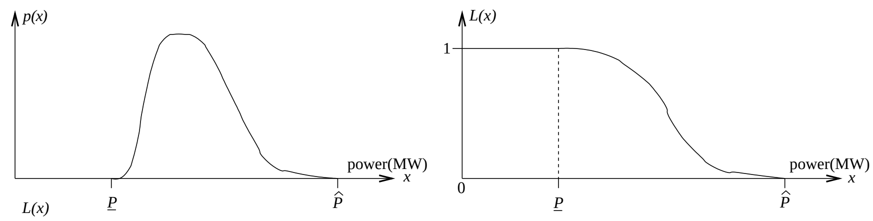

Figure 8.1: Probability density function of load $p(x)$ (left), and cumulative load distribution function $L_0(x)$ (right).

The method calculates the production of each thermal unit, given a loading order. The load is modeled through its distribution $L_0(x)$, which is the probability of requiring x MW, or more. Let:

U_j	:	set of unit indices $1, 2, \ldots, j$
$L_{U_j}(x)$	:	probability distribution of load still to be matched after loading units $1, 2, ..., j-1, j$
x	:	load in MW

the convolution computes $L_{U_j}(x)$ from $L_{U_{j-1}}(x)$ as [1, 15]:

$$L_{U_j}(x) = q_j\, L_{U_{j-1}}(x) + (1 - q_j)\, L_{U_{j-1}}(x + C_j) \tag{8.1}$$

Recalling that $E{=}P{\cdot}T$, the expected energy generated by unit j is [1]:

$$E_j = (1 - q_j)\, T \int_0^{C_j} L_{U_{j-1}}(x)\, dx\ . \tag{8.2}$$

4.2 Unsupplied load after a set of thermal units is loaded

Let $L_0(x)$ be the cumulative probability distribution of the power load corresponding to the LDC. It is not difficult to derive that, given a set of units whose indices 1,2, etc. are the elements of the set of indices Ω, the unsupplied load after loading all the units in Ω will have a cumulative probability distribution $L_\Omega(x)$

$$L_\Omega(x) = L_0(x) \prod_{m\in\Omega} q_m + \sum_{U\subseteq\Omega} \big(L_0(x + \sum_{i\in U} C_i) \prod_{i\in U}(1-q_i) \prod_{i\in\Omega\setminus U} q_i\big) \tag{8.3}$$

We can thus say that the cumulative probability distribution $L_\Omega(x)$ of the unsupplied load is the same no matter the order in which the units in Ω have been loaded.

The unsupplied energy $W(\Omega)$ is computed as:

$$W(\Omega) = T \int_0^{\widehat{P}} L_\Omega(x)\, dx \tag{8.4}$$

The integration in (8.4) is to be carried out numerically.

5 Bloom & Gallant's model for matching the load-duration curve when there are non-load-matching constraints

Let the Bloom & Gallant formulation (for a single interval) [2] be given by:

$$\begin{aligned}
\underset{E_j}{\text{minimize}} \quad & \sum_{j=1}^{n_u+1} \widetilde{f}_j\, E_j && (8.5)\\
\text{subject to} \quad & \sum_{j\in U} E_j \le \widehat{E} - W(U) \quad \forall\, U \subset \Omega = \{1,\dots,n_u\} && (8.6)\\
& A_{\ge}\, E \ge R_{\ge} && (8.7)\\
& A_{=}\, E = R_{=} && (8.8)\\
& \sum_{j=1}^{n_u+1} E_j = \widehat{E} && (8.9)\\
& E_j \ge 0 \qquad j = 1,\dots,n_u, n_u+1 && (8.10)
\end{aligned}$$

where:

n_u+1 : index representing the external energy
$n_\geq$: total number of non-load-matching inequality constraints
$A_\geq$: $\in \mathbb{R}^{n_\geq \times n_u}$ matrix of non-load-matching inequality constraints
$R_\geq$: *rhs* of non-load-matching inequality constraints
$A_=$: $\in \mathbb{R}^{n_= \times n_u}$ matrix of non-load-matching equality constraints
$R_=$: *rhs* of non-load-matching equality constraints
U : subset of Ω
$W(U)$: unsupplied energy after loading all units $j \in U \subset \Omega$

The objective function (8.5) can be simplified using (8.9), which leads to:

$$\sum_{j=1}^{n_u} f_j\, E_j + \widetilde{f}_{n_u+1}\widehat{E} \qquad \text{where} \quad f_j = \widetilde{f}_j - \widetilde{f}_{n_u+1}$$

with $\widetilde{f}_{n_u+1}\widehat{E}$ being a constant.

5.1 The case where no constraint (8.7) is active

Constraints (8.7) and (8.8) are the non-load-matching constraints. The Appendix of [8] contains a proof that the merit-order loading energies correspond to a minimum of the formulation (8.5–8.10) when there are no active constraints (8.7) and in case that there should be no non-load-matching equality constraints (8.8).

Assuming that units are ordered in order of merit, the active constraints at the minimizer of the set of inequalities (8.6) would be:

$$\begin{aligned} E_1 = &\quad \widehat{E} - W(1) \\ E_1 + E_2 = &\quad \widehat{E} - W(1,2) \\ \cdots & \\ E_1 + E_2 + E_3 + \cdots + E_{n_u} = &\quad \widehat{E} - W(1,2,\ldots,n_u) \end{aligned} \tag{8.11}$$

5.2 The case with equalities (8.8) or where a constraint (8.7) or nonnegativity bound (8.10) is active

In this case, the equalities (8.8) or at least one of the constraints in (8.7) or nonnegativity bound (8.10) will be active, which means that at least one of the active constraints in (8.11) will not be satisfied as an equality.

5.3 The multi-interval Bloom and Gallant model

As power planning for a long time period cannot take into account changes over time of some parameters, the time period is subdivided into shorter *intervals* in which all parameters can be assumed to be constant. We will use superscript ^{i} to indicate that variables and parameters refer to the $i^{\underline{\text{th}}}$ interval.

Therefore some constraints refer only to variables of a single interval, while others may refer to variables in several intervals. E.g., constraints on the minimum consumption of gas may affect several or all the intervals, while emission limit constraints, or the constraint associated with the units composing a combined-cycle unit refer to each single interval.

The overhauling of thermal units must be taken into account. Therefore, there will be intervals where some units must remain idle. The set of available units in each interval may be different. Let Ω^i be the set of available units in the $i^{\underline{\text{th}}}$ interval, and let n_u^i be n_u^i=$|\Omega^i|$ (the cardinality of this set).

The Bloom and Gallant linear optimization model extended to n_i intervals, with inequality and equality non-load-matching constraints, can thus be expressed as:

$$\underset{Ej^i}{\text{minimize}} \quad \sum_{i=1}^{n_i}\sum_{j=1}^{n_u} f_j E_j^i \tag{8.12}$$

$$\text{subject to:} \quad \sum_{j\in U} E_j^i \leq \widehat{E}^i - W^i(U) \quad \forall U \subset \Omega^i \quad i=1,\ldots,n_i \tag{8.13}$$

$$A_{\geq}^i\, E^i \geq R_{\geq}^i \qquad i=1,\ldots,n_i \tag{8.14}$$

$$\sum_i A_{\geq}^{0i}\, E^i \geq R_{\geq}^0 \tag{8.15}$$

$$A_{=}^i\, E^i = R_{=}^i \qquad i=1,\ldots,n_i \tag{8.16}$$

$$\sum_i A_{=}^{0i}\, E^i = R_{=}^0 \tag{8.17}$$

$$E_j^i \geq \underline{0} \qquad j=1,\ldots,n_u, \quad i=1,\ldots,n_i \tag{8.18}$$

where:

$A_{\geq}^i$: $\in \mathbb{R}^{n_{\geq}^i \times n_u}$ matrix of inequalities that refer only to interval i
$A_{\geq}^{0i}$: $\in \mathbb{R}^{n_{\geq}^0 \times n_u}$ matrix of inequalities that refer to more than one interval
$R_{\geq}^i$: $\in \mathbb{R}^{n_{\geq}^i}$ *rhs* of inequalities that refer only to energies of interval i
$R_{\geq}^0$: $\in \mathbb{R}^{n_{\geq}^0}$ *rhs* of inequalities that refer to more than one interval
$A_{=}^i$: $\in \mathbb{R}^{n_{=}^i \times n_u}$ matrix of equalities that refer only to energies of interval i
$A_{=}^{0i}$: $\in \mathbb{R}^{n_{=}^0 \times n_u}$ matrix of equalities that refer to more than one interval
$R_{=}^i$: $\in \mathbb{R}^{n_{=}^i}$ *rhs* of equalities that refer only to energies of interval i
$R_{=}^0$: $\in \mathbb{R}^{n_{=}^0}$ *rhs* of equalities that refer to energies of more than one interval

The number of variables is now $\sum_i^{n_i} n_u^i$ and there are $\sum_i^{n_i}(2^{n_u^i}-1)$ load-matching constraints plus $n_= = n_=^0 + \sum_i n_=^i$ non-load-matching equalities, and $n_\geq = n_\geq^0 + \sum_i n_\geq^i$ non-load-matching inequalities. Note that supraindices 0 indicate constraints which affect variables of more than one interval.

Should constraint sets (8.15) and (8.17), which are the multi-interval constraints, be empty, the problem would be separable into n_i subproblems, one for each interval. Otherwise a joint solution must be found.

5.4 Approximate model of long-term hydrogeneration

The long term model described is appropriate for thermal generation units but not for hydrogeneration, which requires additional variables to represent the variability of water storage in reservoirs and discharges necessary for the calculation of the hydroenergy generated.

A coarse model of hydrogeneration, which does not consider any of the reservoir dynamics, can be employed. All or a part of the reservoir systems of one or several basins are considered as a single pseudo-thermal unit H with cost $\widetilde{f}_H=0$, outage probability $q_H=0$ and capacity C_H (normally lower than the maximum installed hydropower capacity), with a constraint binding the intervals' hydrogenerations over the successive intervals so that they add up to a total expected hydrogeneration R_H^0 for the whole period:

$$\sum_i^{n_i} E_H^i = R_H^0 \,, \tag{8.19}$$

6 Long-term maximization of profit in a "competitive" market

In the classical electricity markets, utility companies have both generation and distribution of power. These companies have their own load to supply, corresponding to their clients plus other contracts, and try to minimize their generation cost. In "competitive" electricity markets, generation companies have no distribution, and therefore no load of their own. Generation companies must bid their generation to the market operator and a market price is determined for each hour by matching the demand with the generation of the lowest bids. Generation companies are no longer interested in generating at the lowest cost but in obtaining the maximum profit, which is the difference between market price and generation cost for all accepted generation bids. In long-term operation all accepted bids in a time interval (a week, or a month) must match the LDC of this interval.

There is no specific load to be matched by a specific generation company (SGC). The only known loads are the predicted LDC's for the whole market in each interval. As all generation companies pursue their maximum profit, it is natural to attempt to maximize the profit of all generation companies combined.

The SGC must thus solve the problem of the maximization of profit of all generation companies, taking into account the total market load. The SGC should introduce its own operation constraints (fuel and emission limits, contracts, etc.) and may also introduce a market-share constraint for its units in one or several intervals. (The Lagrange multiplier value of this constraint will tell whether the market share imposed, though feasible, is reasonable or not.) The long-term results will indicate how the SGC should program its units so that its profit be maximized while meeting all its operation constraints.

6.1 Long-term market price function of a given interval

From the records of past market-price and load series (see Fig. 8.2) it is possible to compute a market-price function for a given interval. This function is to be used with expected generations that match the LDC of the interval, so market prices should correspond in duration with the duration of loads, from peak to base load in the interval.

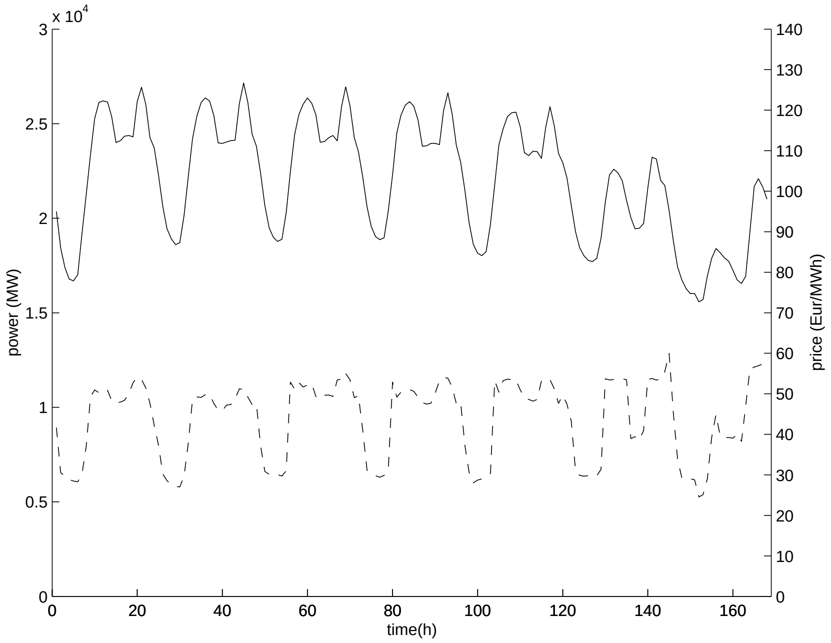

Figure 8.2: Hourly loads (continuous curve) and market prices (dashed)

Both the load and the market price series should be reordered in decreasing load order obtaining a LDC and a price-duration curve that corresponds to the loads in the LDC. The price-duration-curve obtained will be nonsmooth and may even be nondecreasing (see Fig. 8.3). However, fitting a straight line or a low order polynomial to it, a decreasing line or function will generally be obtained. Given the variability of the price-duration curve, it seems reasonable to fit a straight line to it. Let b^i and l^i be the basic and linear coefficient of such line for the $i^{\underline{\text{th}}}$ interval. (Predictions of b^i and l^i could be obtained taking into account both the series corresponding to the same interval in several successive years and that of successive intervals.)

6.2 Maximum profit objective function

In order to determine the maximum-profit objective function, a simplifying assumption is convenient regarding the shape of the unit contributions in the generation-duration curve. Instead of having some units (particulary those with the lowest loading order) with an irregular shape in its right side, it will be assumed that the contribution of all units will have a rectangular shape with height C_j (for unit j) and base length E^i_j/C_j as in Fig. 8.4.

The profit (price minus cost) of unit j in interval i will be:

$$\int_0^{E^i_j/C_j} C_j\{b^i + l^i t - \widetilde{f}_j\}\mathrm{d}t = \left(b^i - \widetilde{f}_j\right)E^i_j + \frac{l^i}{2C_j}{E^i_j}^2$$

and adding for all intervals and units, taking into account the external energy and using (8.9) we get the profit function to be maximized:

$$\sum_i^{n_i}\left[\sum_j^{n_u}\left\{\left(b^i - f_j\right)E^i_j + \frac{l^i}{2C_j}{E^i_j}^2\right\} - \widetilde{f}_{n_u+1}\widehat{E}^i\right] \tag{8.20}$$

with $f_j{=}\widetilde{f}_j{-}\widetilde{f}_{n_u+1}$, which is quadratic in the generated energies. Given that

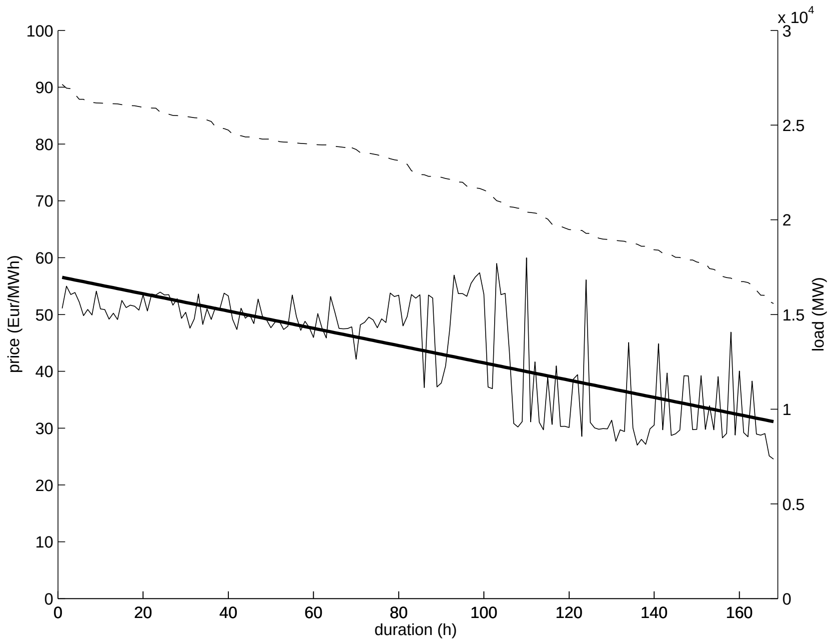

Figure 8.3: Market prices ordered by decreasing load power (thin continuous curve) in weekly interval, market-price linear function (thick line), and LDC (dashed).

$f_{n_u+1}\widehat{E}^i$ is a constant, the problem to be solved is:

$$\underset{Ej^i}{\text{minimize}} \quad \sum_i^{n_i}\sum_j^{n_u}\left\{\left(f_j - b^i\right)E_j^i - \frac{l^i}{2C_j}E_j^{i\ 2}\right\} \tag{8.21}$$

$$\text{subject to:} \quad \sum_{j\in U} E_j^i \leq \widehat{E}^i - W^i(U) \quad \forall U \subset \Omega^i \quad i = 1,\ldots,n_i \tag{8.22}$$

$$A_{\geq}^i\, E^i \geq R_{\geq}^i \qquad i = 1,\ldots,n_i \tag{8.23}$$

$$\sum_i A_{\geq}^{0i}\, E^i \geq R_{\geq}^0 \tag{8.24}$$

$$A_{=}^i\, E^i = R_{=}^i \qquad i = 1,\ldots,n_i \tag{8.25}$$

$$\sum_i A_{=}^{0i}\, E^i = R_{=}^0 \tag{8.26}$$

$$E_j^i \geq \underline{0} \qquad j = 1,\ldots,n_u, \quad i = 1,\ldots,n_i \tag{8.27}$$

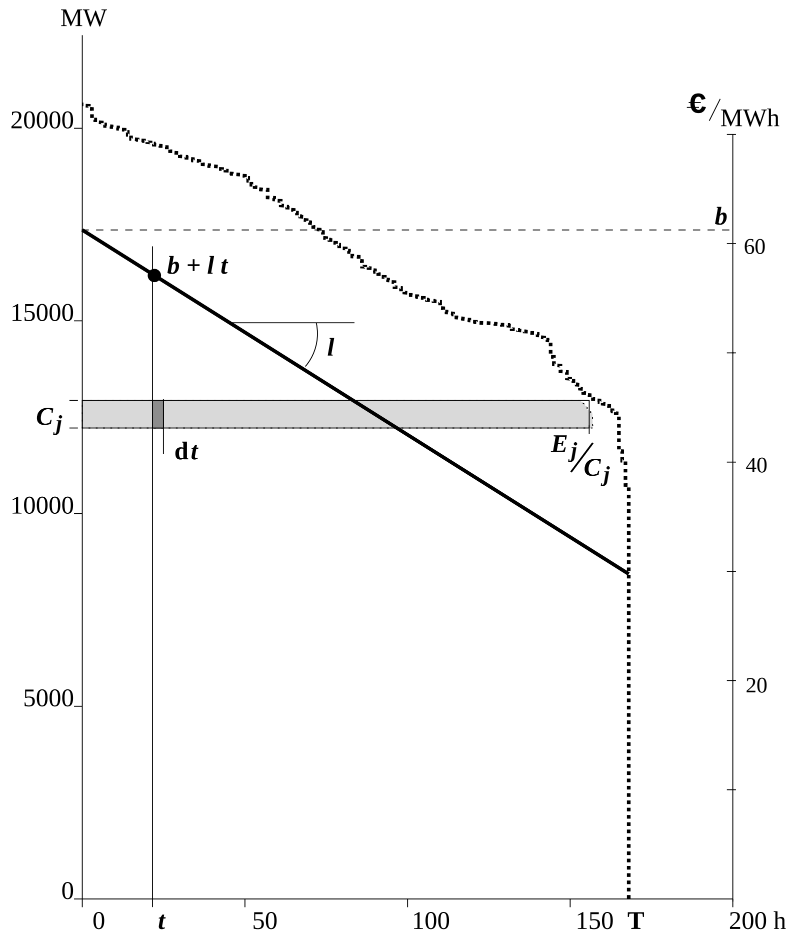

Figure 8.4: Long-term price function for a time interval and contribution of $j^{\underline{\text{th}}}$ unit.

It should be noted that, should all b^i and l^i be zero, the solution of the maximum profit problem (8.24) would be the same as that of the minimum cost problem (8.12-8.18). Otherwise, the cost of the maximum profit solution is higher than that of the minimum cost solution.

Given that $l^i\!<\!0$, the quadratic of the objective function of (8.24) is positive definite, thus problem (8.24) has a unique global minimizer.

7 Coding the load-matching constraints

The main difficulty of the direct solution of the Bloom and Gallant model is the exponential number of load-matching inequality constraints (8.13). These constraints are avoided in the application of the Ford-Fulkerson [13] or the Dantzig-Wolfe column generation method [14, 12, 13], or are generated as they are required in the active set method [9]. In a direct solution by linear or quadratic programming all $n_i \times (2^{n_u} - 1)$ constraints must be explicitly created.

Leaving aside the storage and processing time for these many load-matching inequality constraints, their creation has two parts: the linear coefficients, which is fast [10], and the *rhs*'s, which is very time consuming as it requires lots of calculation.

For each interval i and for the units of each subset U of the set Ω^i we must first calculate $L^i_U(x)$ starting from $L^i_0(x)$ by successive convolution for all units j in U using (8.1), and then compute

$$\widehat{E}^i - W^i(U) = \widehat{E}^i - T^i \int_0^{\widehat{P}^i} L^i_U(x) \mathrm{d}x$$

using numerical integration. This means a lot of arithmetic operations.

8 The Ford-Fulkerson column-generation method applied to the multi-interval problem

Constraints (8.22) and (8.27) define, for each interval, a convex polyhedron whose vertices can be easily calculated. To apply the Ford-Fulkerson procedure, energies $E^i \in \mathrm{I\!R}^{n_u}$ must be expressed as convex combinations of all vertices V^i_k of the $i^{\underline{\mathrm{th}}}$ interval polyhedron:

$$E^i = V^i \Lambda^i \,, \quad V^i \in \mathrm{I\!R}^{n_u \times n^i_V} \qquad \Lambda^i \geq \underline{0}, \qquad \mathrm{I\!I}' \Lambda^i = 1 \quad \forall\, i$$

$\mathrm{I\!I}' = [\; 1 \quad 1 \quad \dots \quad 1 \;]$ being the all one vector.

The number n^i_V of vertices of one such polyhedron is very high as the number of constraints (8.22) that define it, jointly with the nonnegativity bounds (8.27), is exponential: 2^{n_u} (which is over a million for n_u=20). Note that no account is made of extreme-rays as the nature of the constraints and nonnegativity bounds prevents these.

Subtracting surpluses $S^i \in \mathrm{I\!R}^{n^i}_{\geq}$, i=$0, 1, \dots, n_i$ in the inequalities, problem

(8.21-8.27) can be rewritten as:

$$\underset{S^0,\ S^i,\ \Lambda^i}{\text{minimize}} \quad \sum_{i=1}^{n_i} \left\{ (f-b^i)' V^i \Lambda^i + \frac{1}{2} \Lambda^{i'} V^{i'} Q^i V^i \Lambda^i \right\} \tag{8.28}$$

$$\text{subject to:} \quad \mathbb{I}' \Lambda^i = 1 \quad i = 1, \ldots, n_i \tag{8.29}$$

$$\left.\begin{array}{l} A^i_= V^i \Lambda^i = R^i_= \\ A^i_\geq V^i \Lambda^i - S^i = R^i_\geq \end{array}\right\} \quad i = 1, \ldots, n_i \tag{8.30}$$

$$\sum_{i=1}^{n_i} A^{0\,i}_= V^i \Lambda^i = R^0_= \tag{8.31}$$

$$\sum_{i=1}^{n_i} A^{0\,i}_\geq V^i \Lambda^i - S^0 = R^0_\geq \tag{8.32}$$

$$S^0 \geq \underline{0} \qquad S^i \geq \underline{0}, \quad \Lambda^i \geq \underline{0} \quad i = 1, \ldots, n_i\,. \tag{8.33}$$

which is quadratic in Λ^i and lends itself to being solved by the column-generating method of Ford-Fulkerson [5].

The convex coefficients $\Lambda^i \in \mathbb{R}^{n^i_V}$, $i=1, \ldots, n_i$ and the surpluses $S^i \in \mathbb{R}^{n^i_\geq}$, $i=0, 1, \ldots, n_i$ of the inequalities are the variables in the problem. In (8.29-8.33) there are linear equality constraints and non-negativity bounds only.

9 Murtagh and Saunders algorithm using a Column Generation procedure

Given a problem such as (8.28-8.33) we apply the Murtagh and Saunders algorithm [7] using the column generation procedure. The outline of the method is:

0.- $k := 0$; Given an initial feasible point Λ_0, S_0, classify each variable as basic, superbasic or nonbasic. Let n_U be the number of superbasic variables.

1.- Compute the projected gradient, $\|Z'G\|$

2.- If $\|Z'G\| \leq \epsilon$

- · Compute the Lagrange multipliers Σ of the active non-negativity bounds.
- · Look for a constraint c_l, having a negative multiplier, $\sigma_l < 0$
- · If there is any $\sigma_l < 0$ then
  - $n_U := n_U + 1$
  - Update $\|Z'G\|$

else

END

3.- If $\|Z'G\| > \epsilon$

· Compute a descent direction for the basic and superbasic variables, d_k

· Determine the step length, α_k

· Update the variables:

- $\Lambda_{k+1} := \Lambda_k + \alpha_k d_{\Lambda_k}$
- $S_{k+1} := S_k + \alpha_k d_{S_k}$

· Update the basic, superbasic and nonbasic sets

· $k := k + 1$; go to step 1.-

9.1 Obtaining an initial feasible point

Obtaining a feasible point is not trivial when there are non-load-matching constraints.

As with the active set methodology [9], the feasible point is obtained from a point satisfying only the load-matching constraints of all intervals and adding one constraint at a time, plus either a non-zero surplus for the constraint added or a new vertex, until all constraints are satisfied. The details of this process can be found in [12].

9.2 Variable classification

In an active set methodology (such as Murtagh and Saunders is), the active constraints at a feasible point Λ_k, S_k are either general linear constraints or simple bounds.

At a typical iteration, the matrix of active constraints $\widehat{A}$ will contain all the general linear constraints and an additional set of rows of the identity matrix that corresponds to variables at zero.

It is important to mention that in this problem there are only lower (non-negativity) bounds because the upper bound 1 for the convex coefficients λ_k^i is implicit in the convexity constraints ($\mathbb{I}'\Lambda^i = 1\ \forall i$).

As we are in quadratic programming, there is no *a priori* number of fixed variables. Let n_N denote the number of fixed (=0) variables at the current iteration. Then the constraints matrix is (conceptually) partitioned as follows

The $n_B \times n_B$ (where $n_B = n_= + n_\geq + n_i$) "basis" matrix B is square and non-singular, and its columns correspond to the *basic* variables. The n_N columns of N correspond to the *nonbasic* variables (those fixed at 0). The $n_U = (n_V + n_\geq) - n_B - n_N$ columns of the U matrix correspond to the remaining variables, which will be termed *superbasic*.

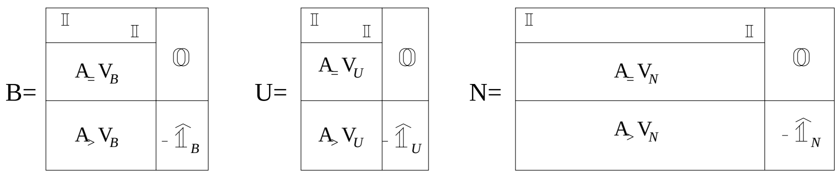

Figure 8.5: General constraint matrix partitioned into basic, superbasic and nonbasic matrices

9.3 The projected gradient

A necessary (but not sufficient) condition to be at the optimizer is that the projected gradient vanishes:

$$\|Z'G\| \leq \epsilon$$

We define the matrix of the null space Z, such that $\widehat{A}Z = 0$, as

$$Z = \left[\begin{array}{c} -B^{-1}U \\ \hline \mathbb{1} \\ \hline \mathbb{0} \end{array} \right]$$

$\mathbb{1}$ is the identity matrix of size n_U, number of superbasics.

G is the gradient of the objective function. As we are dealing with a quadratic function, the gradient at the point Λ_k, S_k for each group of variables is:

$$\begin{array}{ll} \cdot\ G_{\Lambda_B} = (f-b)'V_B + V_B'QV_B\Lambda_B + V_B'QV_U\Lambda_U + V_B'QV_N\Lambda_N & G_{S_B} = 0 \\ \cdot\ G_{\Lambda_U} = (f-b)'V_U + V_U'QV_B\Lambda_B + V_U'QV_U\Lambda_U + V_U'QV_N\Lambda_N & G_{S_U} = 0 \\ \cdot\ G_{\Lambda_N} = (f-b)'V_N + V_N'QV_B\Lambda_B + V_N'QV_U\Lambda_U + V_N'QV_N\Lambda_N & G_{S_N} = 0 \end{array}$$

The terms where Λ_N appears, vanish because $\Lambda_N = 0$. The final expression of the projected gradient is as follows:

$$Z'G = G_U - U'B^{-1'}G_B = G_U - U'\Pi \qquad (8.34)$$

where G_B and G_U refer to the gradient with respect to the basic and superbasic, G_N to that of the nonbasic, and Π comes from solving system $B'\Pi = G_B$

9.4 Computation of the multipliers and generation of new vertices

The overdetermined system $\widehat{A}' \left[\begin{array}{c} \Pi \\ \Sigma \end{array} \right] = G$ is compatible when $Z'G = 0$. The detailed subsystem $\left[\begin{array}{cc} B' & \\ N' & \mathbb{1} \end{array} \right] \left[\begin{array}{c} \Pi \\ \Sigma \end{array} \right] = \left[\begin{array}{c} G_B \\ G_N \end{array} \right]$ is:

$$\left[\begin{array}{c|c|c|c} \mathbb{I} & (A_{=}V_B)' & (A_{\geq}V_B)' & \mathbf{0} \\ \hline \multicolumn{2}{c|}{\mathbf{0}} & -\widehat{\mathbf{1}}_B & \\ \hline \mathbb{I} & (A_{=}V_N)' & (A_{\geq}V_N)' & \mathbf{1} \\ \hline \multicolumn{2}{c|}{\mathbf{0}} & -\widehat{\mathbf{1}}_N & \end{array}\right] \left[\begin{array}{c} \pi_\Lambda \\ \hline \pi_{A=} \\ \pi_{A\geq} \\ \hline \Sigma \end{array}\right] = \left[\begin{array}{c} G_B \\ \hline G_N \end{array}\right] \tag{8.35}$$

The solution procedure solves first

$$B' \left[\begin{array}{c} \Pi_\Lambda \\ \Pi_A \end{array}\right] = G_B$$

for $\Pi_\Lambda \in \mathbb{R}^{n_i}$, which are the multipliers of the convexity constraints, and $\Pi_A \in \mathbb{R}^{n_{=}+n_{\geq}}$, which are the multipliers of the non-load-matching constraints. This calculation is already performed in the projected gradient computation (8.34).

From the equations that yield the multipliers Σ two possible types of equation follow. Either:

$$\pi_\lambda^i + v_{Nk}^{i'}(A'\Pi_A)^i + \sigma_k^i = G_{Nk}^i \qquad i = 1, \ldots, n_i \tag{8.36}$$

hence (recall that $G_{Nk}^i = (f - b^i)'v_{Nk}^i + v_{Nk}^{i'}Q^iV_B^i\Lambda_B^i + v_{Nk}^{i'}Q^iV_U^i\Lambda_U^i$):

$$\sigma_k^i = ((f - b^i) + Q^iV_B^i\Lambda_B^i + Q^iV_U^i\Lambda_U^i - (A'\Pi_A)^i)'v_{Nk}^i - \pi_\lambda^i \tag{8.37}$$

i.e., there is a nonbasic vertex of interval i if $\sigma_k^i<0$, and this will be so if for the modified costs $\widehat{f}^i = (f-b^i)+Q^iV_B^i\Lambda_B^i+Q^iV_U^i\Lambda_U^i-(A'\Pi_A)^i$ the vector of energies v_{Nk}^i yields a cost lower than π_λ^i.

The other equation we obtain from the Σ equations is:

$$-\pi_{\geq k}^i + \sigma_k^i = 0 \quad \rightarrow \quad \sigma_k^i = \pi_{\geq k}^i \qquad i = 0, 1, \ldots, n_i \tag{8.38}$$

which tells that the surplus $s_{\geq k}^i$ $i{=}1, 2, \ldots, n_i, 0$ will become superbasic (relaxing the active constraint $A_{\geq k}^i$) whenever $\pi_{A\geq k}^i<0$.

The problem of finding a (nonbasic) vertex v_{Nk}^i appears to be simple, because given $\widehat{f}^i$ it is straightforward to sort the elements of $\widehat{f}^i$ in increasing value determining a loading order, and compute the elements of v_{Nk}^i by successive convolution

(8.1) and integration (8.2). (And checking whether $\widehat{f^i}' v^i_{Nk} < \pi^i_\lambda$ for some interval i.)

It is in the calculation of vertices that the nonavailability of units — by programmed overhauling during the interval — is taken into account.

9.5 Finding a descent direction

If we have not reached the optimizer, we must find another feasible point that decreases the objective function value. As we are dealing with a constrained problem, a feasible direction is $d = Zp_z$, for any p_z:

$$d = \begin{bmatrix} -B^{-1}U \\ \mathbb{1} \\ \mathbb{0} \end{bmatrix} p_z = \begin{bmatrix} -B^{-1}Up_z \\ p_z \\ \mathbb{0} \end{bmatrix} = \begin{bmatrix} d_B \\ d_U \\ d_N \end{bmatrix}$$

where the nonbasic variables do not change their value. The projected gradient direction, $p_z = -Z'G'$, can be employed or Newton's method:

$$Z'HZp_z = -Z'G' \tag{8.39}$$

where $H = V'QV$ is the Hessian matrix.

Our computational experience is that the projected gradient direction has a poor convergence. Newton's direction is computationally harder to obtain but is much more efficient. When the step length applied is 1, only one iteration is required to achieve $\|Z'G'\| < \epsilon$. However, the computational experience shows that, when applying a step length of 1 using Newton's direction, sometimes rounding errors make necessary more than one iteration.

9.6 Computation of the step length

Given a feasible point Λ_k, S_k and a direction d_k, we choose a new point as $\Lambda_{k+1} := \Lambda_k + \alpha_k d_{\Lambda_k}$ and $S_{k+1} := S_k + \alpha_k d_{S_k}$ where d_{Λ_k} and d_{S_k} are the components of d_k related to Λ_k and S_k respectively. The optimal step length

$$\alpha^*_k = \frac{-G'_{\Lambda_k} d_{\Lambda_k}}{d'_{\Lambda_k} V'QV d_{\Lambda_k}}$$

should be $\alpha^*_k = 1$ if we use Newton's direction. α^*_k may lay beyond the upper limits due to the basic and superbasic variable change.

The variables must be nonnegative, thus:

$$\overline{\alpha}_{Bk} = \text{mín}\left\{ \frac{\lambda^j_{Bk}}{|d^j_{\Lambda_{Bk}}|} \; \forall j \mid d^j_{\Lambda_{Bk}} < 0 \; , \frac{S^j_{Bk}}{|d^j_{S_{Bk}}|} \; \forall j \mid d^j_{S_{Bk}} < 0 \right\}$$

$$\overline{\alpha}_{Uk} = \text{mín}\left\{ \frac{\lambda^j_{Uk}}{|d^j_{\Lambda_{Uk}}|} \; \forall j \mid d^j_{\Lambda_{Uk}} < 0, \frac{S^j_{Uk}}{|d^j_{S_{Uk}}|} \; \forall j \mid d^j_{S_{Uk}} < 0 \right\}$$

The step length is $\alpha_k = \text{mín}\{\overline{\alpha}_{Bk}, \overline{\alpha}_{Uk}, \alpha_k^*\}$. Depending on which one gives α_k, changes in the basic, superbasic and nonbasic sets may occur.

9.7 Changes in the variable sets

Should α_k be $\alpha_k = \alpha_k^*$, no changes occur in the working set. Should α_k be $\alpha_k = \overline{\alpha}_{Uk}$, a superbasic variables becomes zero and changes to nonbasic.

The case of $\alpha_k = \overline{\alpha}_{Bk}$ is more complicated because a basic variable becomes zero, and changes to nonbasic while a superbasic variable changes to basic to substitute it. The new basis has to be refactorized.

In theory, the superbasic chosen to be basic has only to be linearly independent from the remaining basics. In practice, sometimes we can get stuck without any apparent reason.

9.8 Choosing a superbasic variable to enter the basis

The choice of superbasic to enter the basis is important for the solution accuracy and convergence. It is convenient to keep the condition number of B as low as possible in order to get accurate calculations of Π and d_B.

Once known the basic variable l that leaves the basis, which of the superbasic ones will perform better?

The new basis $\widetilde{B}$ will be as the former one B except the leaving column l. The change using an η matrix, can be expressed

$$\widetilde{B} = B\eta$$

whose column different from the unit matrix is in position l.

$$\eta = \begin{pmatrix} 1 & & & w_1 & & \\ & 1 & & w_2 & & \\ & & \ddots & \vdots & & \\ & & & w_l & & \\ & & & \vdots & \ddots & \\ & & & w_{n_B} & & 1 \end{pmatrix}$$

with the components w_i obtained from the vector w that solves $Bw = U_e$. U_e being the entering column of the superbasic set. It is easy to find an upper bound to the condition number of the new basis:

$$cond(\widetilde{B}) \leq cond(B) \cdot cond(\eta)$$

The eigenvalues of η are all ones except w_l, thus its condition number is

$$cond(\eta) = \begin{cases} \text{if } w_l \leq 1 & \rightarrow 1/w_l \\ \text{if } w_l > 1 & \rightarrow w_l \end{cases}$$

The calculation of the lth row of $B^{-1}U$ is at no cost if using Newton's direction, since the explicit calculation of the upper part of Z is required.

9.9 Management of the nonbasic set. Differences between Ford-Fulkerson algorithm and Dantzig-Wolfe algorithm

The main advantage of the column-generation procedure is that vertices (columns) are only generated when they are required. The basic and superbasic vertices have to be generated and stored properly. At the beginning, there is no nonbasic vertex but as the procedure evolves some nonbasic vertices are known. We can do two things with them: get rid of them or store them (and in a next iteration any known nonbasic vertex can become superbasic again).

The version in which the known nonbasic vertexs are deleted is called the Ford-Fulkerson algorithm (FF) and the one that keeps them is called the Dantzig-Wolfe algorithm (DW). In the DW algorithm, before generating a new vertex, the multipliers of the known nonbasic vertices are computed and if there is any negative, it is reentered as a superbasic. In the results section both methods are compared.

10 Computational results

10.1 Test cases

The characteristics of the test cases employed are summarized in Table 8.1. The fourth column, $\sum_i n_u^i$, is the number of variables and the last but two column contains $\sum_i (2^{n_u^i} - 1)$, which is the number of load-matching inequality constraints. All cases except ltp06 correspond to a certain Spanish generation company together with the rest of the Spanish power pool with a different degree of desaggregation of the generation units; the loads satisfied are those of the Spanish power pool. Case ltp06 refers to the planning of a single German generation company considering only its own load. One or more pseudo-units represent, in all cases, the hydrogeneration of one or several basins using the approximate hydromodel of section 5.4.

Market-share constraints can be imposed. Cases whose name ends with "a" do not have any market-share constraint imposed. Cases ending with "b" have market-share constraints associated to the units of the SGC imposed and active.

As mentioned earlier, the purpose of these problems and computational tests is twofold:

- to test the models developed, described in this work, and to observe the influence of several parameters associated with the models, and

Table 8.1: Test cases for long-term electric power planning

case	n_i	n_u	$\sum_i n_u^i$	$\sum_i n_=^i$	$n_=^0$	$\sum_i n_\geq^i$	$n_\geq^0$	$\sum_i(2^{n_u^i}-1)$	solver	$\sum_i \sum_j^{n_u+1} f_j E_j^i$ (€)
ltp01a	11	13	140	0	2	0	2	79861	Cplex	4837512292
ltp01b	11	13	140	0	2	1	4	79861	Cplex	4854704625
ltp02a	11	15	162	0	2	33	3	319477	Cplex	3587429530
ltp02b	11	15	162	0	2	34	5	319477	Cplex	3622023526
ltp03a	11	17	183	0	2	54	5	1245173	Cplex	3580260681
ltp03b	11	17	183	0	2	55	7	1245173	Cplex	3624657306
ltp04a	11	18	193	0	2	64	6	2457589	Cplex	3579624419
ltp04b	11	18	193	0	2	65	8	2457589	Cplex	3624160513
ltp06	15	29	416	0	1	15	3	3758096369	ac.set	1070527267
ltp09a	52	18	931	0	0	312	23	13631436	FFcg	5841845922.26
ltp09b	52	18	931	0	0	313	25	13631436	FFcg	5843484929.38
ltp10a	27	25	658	0	0	162	28	905969637	FFcg	7103008363.89
ltp10b	27	25	658	0	0	163	30	905969637	FFcg	7077804321.51

- to have reliable results (obtained with a reliable code for linear and quadratic programming: *Cplex 7.5*) for the problems posed with which to check alternative specialised algorithms to solve the same problems, specifically the Dantzig-Wolfe and Ford-Fulkerson column generation algorithms, the active set algorithm, and other algorithms to be developed.

10.2 Performance of the Ford-Fulkerson procedure and comparison with the active set method

Both the active set and the FFcg methods require a considerable number of iterations to reach a feasible solution. Their numbers appear under the heading "feas. iters." (feasibility iterations) in Table 8.2; the number of iterations to achieve the optimizer is shown next. After that, the required CPU time, and the number of figures of agreement of the objective function value with that obtained with a different solver are shown, as indicated in the last two columns of Table 8.1. The last three columns in Table 8.2 show the results obtained using an *AMPL* plus *Cplex 7.5* solution [10], the last column giving the long computation times required, in hours(!), to have the *rhs*'s of the $\sum_i(2^{n_u^i}-1)$ load-matching constraints (8.13).

Several conclusions can be drawn from the results of Table 8.2. The first is that the FFcg method is quicker to get to the solution and that the rate of increase of the time required with problem size is lower in the case of FFcf than

Table 8.2: Comparison of the active set, and the Ford-Fulkerson column generation method

	active set method				Ford-Fulkerson cg						Cplex 7.5		
case	feas. iters.	total iters.	time (s)	dig. ag.	feas. iters.	total iters.	time (s)	ver. gen.	ver. opt.	dig. ag.	total iters.	time (s)	*rhs* (h)
ltp01a	193	246	6.6	10	21	79	7.2	147	15	10	781	1.3	0.44
ltp01b	239	312	9.0	9	21	224	16.4	396	18	9	2354	2.35	0.44
ltp02a	450	642	62.5	10	128	357	14.4	254	20	10	3285	11.0	2.28
ltp02b	513	734	80.1	9	128	516	16.1	293	24	10	7646	16.9	2.28
ltp03a	672	964	197.1	10	310	831	20.5	348	23	10	12622	56.8	9.52
ltp03b	781	1096	348.0	9	310	1213	21.6	354	33	9	23213	86.2	9.52
ltp04a	938	1233	508.2	10	400	796	23.7	383	25	9	17447	115.1	19.27
ltp04b	1075	1404	756.6	10	400	1768	38.5	603	45	9	42785	212.0	19.27
ltp06	1803	2646	24.3	–	51	585	5.0	466	31	10	n.a.	n.a.	n.a.

with the active set or the direct linear programming solution.

The next issue is precision. Direct linear programming, the active set method and the FFcg procedure reach practically the same optimizer (the number of agreement digits of these methods' solution is 9 or more for all cases). Four agreement digits would be fairly acceptable from an engineering view-point, given that many data in this problem are approximations or predictions. Therefore it could be thought that the optimization process could be stopped when the objective function does not change in the first five or six figures over a number of iterations. It must be borne in mind that the active set method for a linear program behaves like linear programming, and obtaining the right set of active constraints produces exactly the same optimizer. However, the FFcg procedure generates the optimizer as the convex combination of vertices of the polyhedrons of feasible points (one for each interval in long-term power planning). Thus the calculation of the optimizer, and its objective function value, requires many more arithmetic operations. The column with header "ver. opt." contains the number of vertices at the optimizer. On average, we have 2 vertices for each interval.

All test cases have been solved with two different objective functions: the linear minimum cost (8.12) and the quadratic of maximum profit (8.20). The linear cost problems have been solved using the linear programming code in *Cplex 7.5* package [3], while for the quadratic profit problem the barrier separable QP solver [16] in *Cplex 7.5* package is employed, both through an *AMPL* [6] model and data files. Prior to the solution, the *rhs*'s of the load-matching inequality constraints (8.13) have been calculated using an separate program, whose required CPU time is reported in the last column of Table 8.1. The calculated *rhs*'s are a part of the *AMPL* data files used.

The solutions obtained with the Ford-Fulkerson column generation, which is the most efficient [13], are compared with those obtained through *AMPL* plus *Cplex 7.5* quadratic programming and with Dantzig-Wolfe column generation.

Table 8.3: Comparison of AMPL plus Cplex, and the Dantzig-Wolfe and Ford-Fulkerson column generation methods

	AMPL plus Cplex 7.5				Dantz-Wl.		Ford-Fulk. column gen.		
case	input (s)	b qp iters.	b. qp (s)	obj. fun. (8.21) (€)	D.W. ites.	(s)	F.F. ites.	(s)	obj. fun. (8.21) (€)
ltp01a	1.3	34	97.56	9552335013	289	12.1	262	11.6	9552335013
ltp01b		46	55.09	9536489728	258	9.1	240	9.5	9536489725
ltp02a	5.69	59	183.5	10986157177	842	41.4	629	36.5	10986157163
ltp02b		56	176.9	10961049191	1248	51.8	893	44.1	10961049198
ltp03a	24.47	78	1020.3	11004938184	1321	82.3	957	60.4	11004938185
ltp03b		75	977.7	10977720297	1934	97.4	1341	77.9	10977720295
ltp04a	46.88	87	4393.2	11006374461	1423	91.8	1132	79.1	11006374462
ltp04b		116	5787.0	10979064726	2063	109.0	1545	92.0	10979064723
ltp06					1103	558.3	838	423.4	936301399
ltp09a					7949	24643	7337	18825	5841845922.26
ltp09b					8739	31630	6989	18910	5843484929.38
ltp10a					3666	12392	2831	16412	7103008363.89
ltp10b					5323	19616	4220	9191	7077804321.51

The second column of Table 8.3 has the input times required by the *AMPL* data files. These times are important because the data files, due to the *rhs*'s of the load-matching constraints, are very large, e.g., the data file for case ltp04a is over 100Mbyte.

It can be observed that the Ford-Fulkerson column generation proves to be systematically more efficient in itarations and CPU time than Dantzig-Wolfe's. In the table, the enormous time required to calculate the *rhs* terms of the load-matching constraints when using *AMPL* plus *Cplex 7.5* is not included.

10.3 Solutions of long-term maximum profit planning and comparison with the minimum cost solution

It is clear from the results in Table 8.4 that the maximization of profit with respect to the minimum cost solution brings about a greater increase in generation cost than an increase in profits.

Table 8.4: Minimum cost and maximum profit solutions with an approximate and linearized full hydromodel (using Cplex)

	Minimum expected cost solution				Maximum expected profit solution			
case	lp iters.	lp (s)	cost (8.5) (€)	profit (8.21) (€)	b qp iters.	bar qp (s)	profit (8.21) (€)	cost (8.5) (€)
ltp01a	781	1.3	4837512292	9120093218	34	97.56	9552335013	5414756605
ltp01b	2354	2.35	4854704625	9097863493	46	55.09	9536489728	5427534028
ltp02a	3285	11.00	3587429530	10364887782	59	183.5	10986157177	3971181574
ltp02b	7646	16.92	3622023525	10291956272	56	176.9	10961049191	4018697239
ltp03a	12622	56.85	3580260681	10292023819	78	1020.3	11004938184	3978570531
ltp03b	23213	86.16	3624657306	10287465543	75	977.7	10977720297	4025913005
ltp04a	17220	113.7	3579624419	10306638029	87	4393.2	11006374461	3975489971
ltp04b	42785	212.0	3624160513	10296858318	116	5787.0	10979064726	4025907694

10.4 Effect of market-share constraints

Three market-share constraints have been introduced in cases whose name ends with "b": one for the first interval, one for the intervals corresponding to the rest of the first year (intervals 2 to 7), and a third for the intervals of the second year (8 to 11). These three sets of successive intervals will be referred to with the supraindices ^{I}, ^{II} and ^{III} associated to the variables. The market-share constraints refer to the units of the SGC, and force their generation to add up to over a given percentage of the load in the corresponding intervals.

$$\sum_{i\in Ik}\sum_{j\in SGC} E_j^i \geq \mu^{Ik} \sum_{i\in Ik} \widehat{E}^i \qquad Ik:\ I,\ II,\ III, \tag{8.40}$$

which are of type (8.15), except set I (a single interval) which is of type (8.14).

The criterion employed to fix a market-share μ^{Ik} for the units in the set SGC is based on the Lagrange multiplier values of the market-share constraints λ_{m-s}^{Ik} and the expected profit rate in the power pool r^{Ik}: total profit over total load. The Lagrange multipliers λ_{m-s}^{Ik} express the rate of change in pool profit due to a market-share increase by the SGC. The reaction of competitor generating companies to a market-share increase by the SGC would be proportional to the resulting $\lambda_{m-s}^{Ik}/r^{Ik}$. Therefore, attainable market-shares are those that produce a small enough value $\lambda_{m-s}^{Ik}/r^{Ik}$. In the cases reported in Table 8.5 the market-shares μ^{Ik} of the SGC have been pushed up until the ratio $\lambda_{m-s}^{Ik}/r^{Ik}$ was close to but did not exceed $\frac{1}{3}$.

There are also cases whose name ends with "a" in Table 8.5. These cases are the same as those ending in "b" but without the market-share constraints.

Table 8.5: Effect of market share constraints on the profit of the SGC

case	μ^I %	λ^I_{m-s}	r^I	μ^{II} %	λ^{II}_{m-s}	r^{II}	μ^{III} %	λ^{III}_{m-s}	r^{III}	total profit (€)	SGC profit (€)
ltp01a	3.75	0.0		3.36	0.0		3.44	0.0		9552335013	263380937
ltp01b	4.2	8.25	26.12	4.2	8.93	27.39	4.2	8.24	25.10	9538257985	268453956
ltp02a	1.85	0.0		1.94	0.0		2.2	0.0		10986157177	174506646
ltp02b	3.4	9.64	29.31	3.3	10.24	31.12	3.4	9.45	29.14	10963147542	206313710
ltp03a	2.08	0.0		2.25	0.0		2.57	0.0		11004938184	205156894
ltp03b	3.6	9.58	29.35	3.6	10.14	31.17	3.8	9.64	29.20	10981583153	235282378
ltp04a	2.08	0.0		2.25	0.0		2.59	0.0		11006374461	205051615
ltp04b	3.6	9.35	29.36	3.6	9.84	31.18	3.7	8.94	29.21	10985461774	232575618

They are thus equivalent to having imposed a nonactive market share, lower than the share the SGC gets in the solution. It should be noted that the market-share constraints imposed slightly decrease the overall profit, but they noticeably increase the SGC profit.

11 Conclusions

- The long-term hydrothermal planning of the electricity generation problem has been presented and an extension of the Bloom and Gallant model has been put forward in order to solve it.
- A new way of formulating the long-term profit maximization of generating companies in a competitive market has been described.
- An implementation of the Ford-Fulkerson and of Dantzig-Wolfe column generation procedures for solving a quadratic or a linear problem has been presented.
- Implementation details of the solution with *AMPL* of the minimum cost and the maximum profit long-term planning problems have been given.
- The computational experience with the Ford-Fulkerson and of Dantzig-Wolfe column generation procedures and with *AMPL* plus *Cplex 7.5* linear programming and barrier quadratic programming has been reported. This includes:
  - The calculation of the *rhs*'s of the load-matching constraints for the data files required by *AMPL*, which is extremely time-consuming, and

which is fairly time-consuming to be read in the solution process. This lengthy calculation, requiring extremely long files to store the results, makes this procedure impractical to use for real cases (where the number of units to consider may be well above one hundred).

– The solution of the minimum cost and the maximum profit long-term problems.

– The comparison of the three procedures implemented for a set of real cases, using the approximate hydrogeneration representation, showing that the Ford-Fulkerson column generation is the most efficient except for case *ltp10a*, and that *AMPL* plus *Cplex 7.5* is not practical for big cases.

– The analysis of the effect of market-share constraints for a SGC in the maximum profit solution.

12 Bibliografía

[1] H. Balériaux, E. Jamoulle and F. Linard de Gertechin, (1967). Simulation de l'explotation d'un parc de machines thermiques de production d'électricité couplé à des stations de pompage, *Revue E de la Soc. Royale Belge des Electriciens*, 7: 225–245.

[2] J. A. Bloom and L. Gallant, (1994). Modeling dispatch constraints in production cost simulations based on the equivalent load method, *IEEE Transactions on Power Systems*, 9(2) 598–611.

[3] CPLEX Optimization Inc. *Using the CPLEX Callable Library. User's Manual, version 4.0*, (1985).

[4] G. B. Dantzig and P. Wolfe, (1960). Decomposition principle for linear programmes, *Operations Research*, 8: 101–110.

[5] L. R. Ford and D. R. Fulkerson, (1958). A suggested computation for maximal multicommodity network flows, *Management Science*, 5: 97 - 101.

[6] R. Fourer, D.M. Gay and B.W. Kernighan, (1993). *AMPL a modeling language for mathematical programming*, Boyd and Fraser Publishing Company, Danvers, MA 01293, USA, 1993.

[7] P.E. Gill, W. Murray and M.H. Wright, (1981). Practical Optimization, *Academic Press*, London, G.B.

[8] N. Nabona, and M. A. Díez, (1998). Long-term thermal multi-interval thermal generation scheduling using Bloom and Gallant's linear model, Technical

Report DR9803, Dept. Estadística i Inv. Operativa, Univ. Politècnica de Catalunya, 08028, Barcelona.

[9] N. Nabona, C. Gil, and J. Albrecht, (2001). Long-term thermal power planning at VEW ENERGIE using a multi-interval Bloom and Gallant method, *IEEE Transactions on Power Systems*, 16(1): 69–77.

[10] N. Nabona, and A. Pagès, (2002). Long-term electric power planning in a competitive market using the Bloom and Gallant procedure and a modeling language, Technical Report DR2002/24, Dept. Estadística i Inv. Operativa, Univ. Politècnica de Catalunya, 08028, Barcelona.

[11] A. Pagès, (2002). Modelització amb AMPL de la coordinació tèrmica a llarg termini de la generació elèctrica segons la formulació de Bloom i Gallant, Supervised assignment, Facultat de Matemàtiques i Estadística, Univ. Politècnica de Catalunya, 08028, Barcelona.

[12] A. Pagès, (2002). Modelització amb AMPL de la coordinació tèrmica a llarg termini de la generació elèctrica segons la formulació de Bloom i Gallant. Final-year project, Facultat de Matemàtiques i Estadística, Univ. Politècnica de Catalunya, 08028, Barcelona.

[13] A. Pagès and N. Nabona, (2002). Long-term electric power planning using the Bloom and Gallant's linear model. A comparison of solution techniques, Technical Report DR2002/25, Dept. Estadística i Inv. Operativa, Univ. Politècnica de Catalunya, 08028, Barcelona.

[14] J. Pérez-Ruiz and A. J. Conejo, (2000). Multi-period probabilistic production cost model including dispatch constraints, *IEEE Transactions on Power Systems*, 15(2): 502–507.

[15] A.J. Wood and B.F. Wollenberg, (1984). *Power generation, operation and control*, John Wiley & Sons, New York, USA.

[16] S.J. Wright, (1997). *Primal-dual interior-point methods*, SIAM, Philadelphia, PA 19104-2688, USA.