The active set and the Ford-Fulkerson column generation method in the solution of the long-term thermal power planning using the Bloom and Gallant model.

A. Pagès and N. Nabona
Dept. of Statistics and Operations Research, UPC

DR 2002/25
31–12–2002

Postcript copies of this report may be obtained via anonymous *ftp* at `ftp-eio.upc.es` in directory `pub/onl/reports`. It is the file named `dr200325.ps.Z`. It needs decompressing before printing (e.g. via `uncompress dr200325.ps.Z` to get a postscript file).

corresponding author: Narcís Nabona
Dept. Statistics and Operations Research,
UPC Campus Sud, edifici U,
c. Pau Gargallo 5, 08028 Barcelona,
tel. +34 934017035 fax +34 934015855
email:narcis.nabona@upc.es

The active set and the Ford-Fulkerson column generation method in the solution of the long-term thermal power planning using the Bloom and Gallant model.[1]

Adela Pagès and Narcís Nabona

Abstract: *Bloom and Gallant have proposed an elegant model for finding the optimal thermal schedule subject to matching the load-duration curve and general linear constraints. Their method is based on a linear program with some linear equality constraints and many linear inequality constraints. There have been applications of this procedure to multi-interval problems using the active set method, the Ford-Fulkerson column generation method, and the direct application of linear optimization packages using an available modeling language. This work describes the model of long-term electric power planning adapted to use Bloom and Gallant's procedure and compares the performance of the two first techniques mentioned above to solve a number of long-term power planning problems of various sizes. Several remarks are made on implementation issues of both procedures.*
Keywords: *Linear Programming, Active set optimization methods, Ford-Fulkerson column generation, Long-term power generation scheduling, Stochastic processes, Thermal power generation.*

1 Introduction and motivation

Long-term generation planning is a key issue in the operation of an electric generation company. Its results are used both, for budgeting and planning fuel acquisitions, and to give a frame where to fit 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, e.g., load, availability of thermal units, and hydrogeneration (and generations from renewable sources in general).

A long-term planning *period* (e.g., a natural year) is normally subdivided into shorter time *intervals* (e.g., a week or a month long), for which parameters (e.g., the load-duration curve) are known or predicted, and optimized variables (e.g., 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 (having 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 only thermal units when there are load-matching and other operational non-load-matching constraints such as limits on the availability of certain fuels, or environmental maximum emission limits. The optimal *loading order* obtained by Bloom & 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 placement possible, with respect to generation cost, in the matching of the LDC. (Changes in loading order bring about discrete changes in energy generation, while splittings cause a continuous variation in the energy generated depending on where the split starts [2].)

When the long-term planning power problem is to be solved for a generation company operating in a competitive market, this 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 bids of this and other generating companies to match the load. In this case, the problem scope is no longer that of the generation units of a single generation company but all units of all companies biding in the same competitive market matching the load of the whole system. This makes the size of planning problems much larger than before and is a reason for developing more efficient codes to solve the problem.

[1]Partially supported by CICYT Grant TAP99-1975-C02-01 of *Ministerio de Educación y Cultura*, Spain

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 [5, 12], or the Ford-Fulkerson column-generation (FFcg) method [6, 11]. The FFcg and the Dantzig-Wolfe procedures have many common steps. The differences between them will be discussed later.

In this work, the theoretical bases for the implementation of the FFcg method for a multi-interval long-term power planning problem are explained. The performance of the active-set procedure and the column generation method with a set of power planning problems of different sizes are then compared. The implementation of the obtention of new vertices where some components are zero, or close to zero, is analyzed and some of the computational results reported show its influence.

A companion report to this one by the same authors [10] deals with the third solution procedure mentioned above: the direct application of linear (and nonlinear) optimization packages using an available modeling language, which is only able to solve small problems in a reasonable amount of CPU time, and explains some modeling extensions appropriate for competitive markets [3].

2 The load-duration curve

The LDC is the most sensible way to represent the load of a future interval in an integrated way (the load depends on random factors such as weather in several geographical areas, human decisions, social events, etc.). 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 $\widehat{E}^i$
- ⋆ 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 (see Fig. 1).

It should be noted that in a *predicted* LDC random events such as weather, shifts in consumption timing, etc., that cause modifications of different sign in the load tend to cancel out, and that the LDC conserves all the power variability of the load.

2.1 Power and energy constraints imposed by the load-duration curve

The loading of thermal units in an LDC was first formulated in [1] and practical procedures to compute the covering can be found in [13].

Analytically, given the probability density function of load $p(x)$, the cumulative load distribution function $L_0(x)$ is calculated as:

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

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

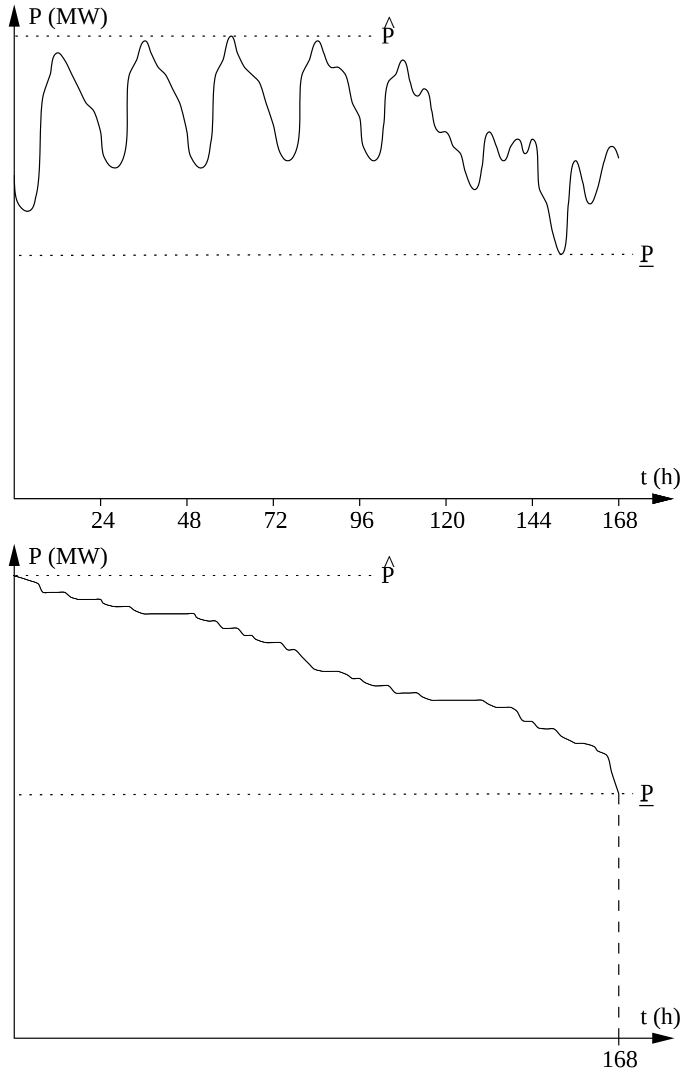

Figure 1: Load over time, above, and load-duration curve (LDC), below . (Data for a week — Monday to Sunday — in January).

⋆ *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 up to its maximum capacity (because the price of its MWh is lower!)

⋆ *loading order*: units will have load allocated to them in a given order

- loading order and merit order should coincide, but when there are other constraints to be satisfied the most economical loading order may be different from the merit order.

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

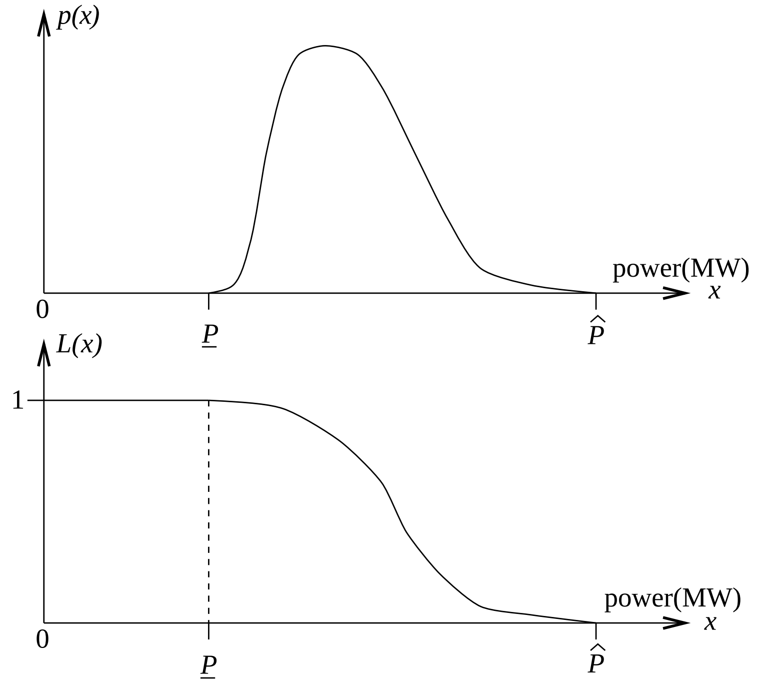

Figure 2: Probability density function of load $p(x)$ (above), and cumulative load distribution function $L_0(x)$ (below).

load: $\sum_{i=1}^{Nu} C_i > \widehat{P}$, and it is normal to find that $\sum_{i=1}^{Nu} C_i \approx 1.4\widehat{P}$..

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 area under the LDC and the area under the generation-duration curve must coincide. (See in Fig. 3 the generation-duration curve corresponding to a given LDC.)

The peak power of the generation-duration curve is $\sum_{i=1}^{Nu} C_i + \widehat{P}$ and the area above power $\sum_{i=1}^{Nu} C_i$ is the external energy. In order to find the generation-duration curve from the LDC, the convolution method of Balériaux, Jamoulle & Linard de Guertechin [1] was implemented.

4.1 Convolution method of finding the generation-duration curve

The method calculates the production of each thermal unit, given a loading order. The load is modeled through its distribution (see Fig. 2):

$$L_0(x) = \begin{cases} 1, & \text{for } x \leq \underline{P} \\ r \in [0,1], & \text{for } \underline{P} < x \leq \widehat{P} \\ 0, & \text{for } x > \widehat{P}. \end{cases}$$

(Recall that $L_0(x)$ is the probability of requiring x MW, or more).

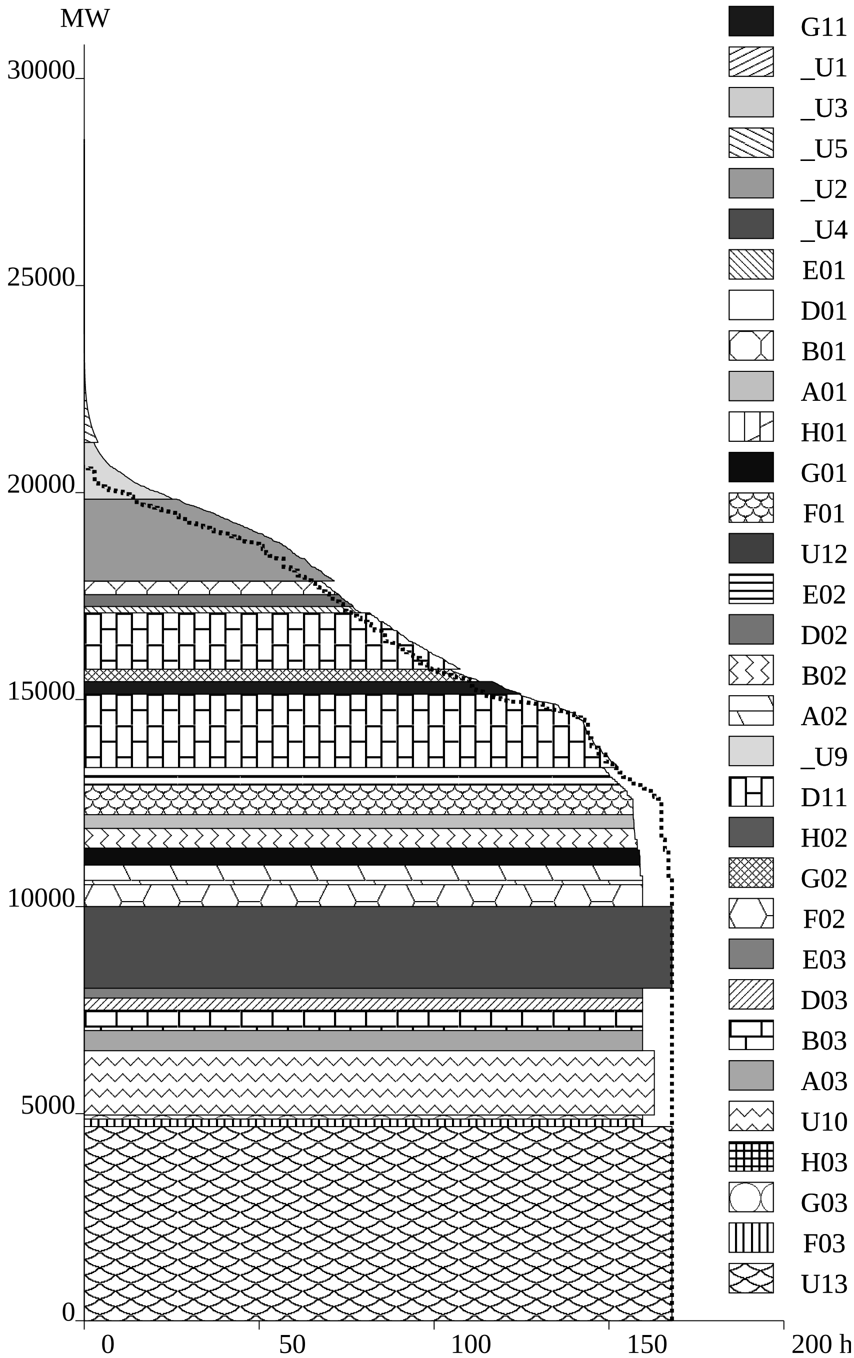

Figure 3: Generation-duration curve and load-duration curve (dotted line) for a weekly interval in a 32 unit problem.

4.2 Convolution of the $j^{\underline{\text{th}}}$ thermal unit

Let:

C_j	:	maximum power capacity in MW of unit j
q_j	:	outage probability of unit j
$1 - q_j$	:	in service probability of unit j
$L_{j-1}(x)$	:	probability distribution of uncovered load after loading units $1, 2, ..., j-1$
$L_j(x)$	:	probability distribution of uncovered load after loading units $1, 2, ..., j-1, j$
x	:	load in MW

the convolution computes $L_j(x)$ from $L_{j-1}(x)$ as [1, 13]:

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

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

$$E_j = (1 - q_j)\, T \int_0^{C_j} L_{j-1}(x)\, dx\ . \tag{2}$$

4.3 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, where x represents load. Once unit i (capacity C_i, and outage probability q_i) has been loaded, the cumulative power distribution of the unsupplied load will be

$$L_{\{i\}}(x) = q_i L_0(x) + (1 - q_i) L_0(x + C_i) \tag{3}$$

If we then load thermal unit j (C_j, q_j) we are left with $L_{U_j}(x)$

$$L_{\{j,i\}}(x) = q_j L_{\{i\}}(x) + (1 - q_j) L_{\{i\}}(x + C_j)$$

which, taking into account (3), can be rewritten as

$$\begin{aligned} L_{\{j,i\}}(x) = \quad & q_i q_j L_0(x) + (1 - q_i) q_j L_0(x + C_i) + (1 - q_j) q_i L_0(x + C_j) \\ & + (1 - q_i)(1 - q_j) L_0(x + C_i + C_j) \end{aligned} \tag{4}$$

Note in (4) that, should we have loaded units i and j in the reverse order, $L_{\{i,j\}}(x)$ (having loaded unit i first and then unit j) would have been the same as $L_{\{j,i\}}j(x)$ (having loaded unit j first and unit i second).

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 $L_\Omega(x)$

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

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.

4.4 Computation of the unsupplied energy

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

$$W(\Omega) = T \int_0^{\widehat{P}} L_\Omega(x)\, dx \tag{5}$$

$L_\Omega(x)$ being the probability distribution of the unsupplied load after loading (in any order) units $j \in \Omega$.

The integration in (5) is to be carried out numerically.

4.5 Loading order and energy generation bounds of a given unit

Let Ω be a set of unit indices. It has been shown before that the distribution $L_\Omega(x)$ is independent of the order in which the units have been loaded. However, the contribution of each unit to matching the load will be different depending on the position in loading order, and the higher the position the bigger the generation will be for the given unit. This is so because of (2) since $L_{j+1}(x) \le L_j(x)\ \forall x$.

A given unit of index $k \in \Omega$ will have its generation bounded by:

$$0 < (1-q_k)\, T \int_0^{C_k} L_{\Omega\backslash k}(x)\, dx = \underline{E}_k \le E_k \le \overline{E}_k = (1-q_k)\, T \int_0^{C_k} L_0(x)\, dx \tag{6}$$

where $L_{\Omega\backslash k}(x)$ corresponds to the probability distribution of uncovered load after loading all units in Ω but that of index k.

$\underline{E}_k$ and $\overline{E}_k$ correspond respectively to loading unit k the last and the first.

5 Bloom & Gallant's modeling of 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:

$$\underset{E_j}{\text{minimize}} \quad \sum_{j=1}^{Nu+1} \widetilde{f}_j\, E_j \tag{7}$$

$$\text{subject to} \quad \sum_{j\in\omega} E_j \le \widehat{E} - W(\omega) \quad \forall\, \omega \subset \Omega = \{1,\ldots,N_u\} \tag{8}$$

$$A_{\ge}\, E_j \ge R_{\ge} \tag{9}$$

$$\sum_{j=1}^{Nu+1} E_j = \widehat{E} \tag{10}$$

$$E_j \ge 0 \qquad j = 1,\ldots,N_u, N_u+1 \tag{11}$$

where:

$Nu+1$	index representing the external energy
$N_{\ge}$	total number of non-load-matching inequality constraints
$A_{\ge} \in \mathbb{R}^{N_{\ge}\times N_u}$	matrix of coefficients of non-load-matching inequality constraints
$R_{\ge}$	*rhs* of non-load-matching inequality constraints
ω	subset of Ω
$W\{\omega\}$	energy not covered after loading all units $j \in \omega \subset \Omega$

The objective function (7) can be simplified using (10), which leads to:

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

Given that $\widetilde{f}_{N_u+1}\widehat{E}$ is a constant, problem (7-11) can be recast as:

$$\begin{aligned} \underset{E_j}{\text{minimize}} \quad & \sum_{j=1}^{N_u} f_j\, E_j \\ \text{subject to} \quad & \sum_{j\in\omega} E_j \le \widehat{E} - W(\omega) \quad \forall\, \omega \subset \Omega = \{1,\ldots,N_u\} \\ & A_{\ge}\, E_j \ge R_{\ge} \\ & E_j \ge 0 \qquad j = 1,\ldots,N_u\,. \end{aligned} \tag{12}$$

5.1 The case where no constraint (9) is active

Constraints (9) 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 (7–11) when there are no active constraints (9).

Assuming that units are ordered in merit order, the active constraints at the minimizer of the set of inequalities (8) would be:

$$
\begin{aligned}
E_1 &= \quad \widehat{E} - W\{\,1\} \\
E_1 + E_2 &= \quad \widehat{E} - W\{\,1,2\} \\
E_1 + E_2 + E_3 &= \quad \widehat{E} - W\{\,1,2,3\} \\
&\cdots \\
E_1 + E_2 + E_3 + \cdots + E_{N_u} &= \quad \widehat{E} - W\{\,1,2,\ldots,N_u\}
\end{aligned}
\tag{13}
$$

Substracting the first from the second equality we get:

$$E_2 = W(1) - W(1,2) > 0$$

and substracting the second from the third:

$$E_3 = W(1,2) - W(1,2,3) > 0\,.$$

Finally, substracting the last but one from the last we would have:

$$E_{N_u} = W(1,2,\ldots,N_u-1) - W(1,2,\ldots,N_u) > 0$$

It must be stressed that all E_i are $E_i{>}0$ because $W(1,2,\ldots,i-1){>}W(1,2,\ldots,i)$ and that however small $W(1,2,\ldots,i-1){-}W(1,2,\ldots,i)$ it may be it will never be zero, i.e., no nonnegativity bound (11) will be active. Moreover, the energies of all units will be within its bounds (6)

The energy generation E_j will coincide with that calculated in (2):

$$
\begin{aligned}
E_j &= W(1,...,j-1) - W(1,...,j-1,j) = T\int_0^{\widehat{P}} L_{j-1}(x)\,\mathrm{d}x - T\int_0^{\widehat{P}} L_j(x)\,\mathrm{d}x \\
&= T\int_0^{\widehat{P}} \big(L_{j-1}(x) - L_j(x)\big)\,\mathrm{d}x = T\int_0^{\widehat{P}} \big(L_{j-1}(x) - q_j L_{j-1}(x) - (1-q_j)L_{j-1}(x+C_j)\big)\,\mathrm{d}x \\
&= T(1-q_j)\int_0^{\widehat{P}} \big(L_{j-1}(x) - L_{j-1}(x+C_j)\big)\,\mathrm{d}x = T(1-q_j)\int_0^{C_j} L_{j-1}(x)\,\mathrm{d}x
\end{aligned}
\tag{14}
$$

5.2 Cases in which a constraint (9) or nonnegativity bound (11) is active

In these cases at least one of the constraints in (9) or nonnegativity bound (11) will be active, which means that at least one of the constraints in (13) is not satisfied as an equality.

Let us assume that j, k, and l are three consecutive units in loading order (which may be different from *merit* order), and that the k^{th} equation of system (13) is not active, while the first equations up to the $j^{\underline{\text{th}}}$ are, and so are the equations from the $l^{\underline{\text{th}}}$ onwards. The values of the energies up to E_j can be obtained by subtracting from each equation its former one, as done in 5.1.

By subtracting the active $j^{\underline{\text{th}}}$ equation from the $l^{\underline{\text{th}}}$ we get:

$$E_k + E_l = W(1,...,j) - W(1,...,j,k,l) \tag{15}$$

The actual value of E_k and E_l will come out as part of the solution of (7–11) and will satisfy (15) and the rest of active constraint, including those of (9) and the nonnegativity bounds (11).

As noted in [2], as regards energies E_k and E_l, the solution can be viewed as a *splitting* of a unit by the other. Let us suppose that unit k is split by unit l, i.e., unit k is decomposed into two independent pseudounits with the same probability of failure q_k, price f_k and capacities $C_{k1}{>}0$ and $C_{k2}{>}0$, such that $C_{k1}+C_{k2}=C_k$, whose corresponding energies will be E_{k1} and E_{k2} ($E_{k1}+E_{k2}=E_k$). The splitting means that the loading order is: unit j, pseudounit $k1$, unit l, pseudounit $k2$. E_{k1} is:

$$E_{k1} = T(1-q_k)\int_0^{C_{k1}} L_j(x)\mathrm{d}x \tag{16}$$

Since L_{k1} is:

$$L_{k1}(x) \;=\; q_k L_j(x) + (1-q_k)L_j(x+C_{k1}),$$

E_l is then:

$$\begin{aligned} E_l &= T(1-q_l)\textstyle\int_0^{C_l} L_{k1}(x)\,\mathrm{d}x \\ &= T(1-q_l)q_k\textstyle\int_0^{C_l} L_j(x)\,\mathrm{d}x + T(1-q_l)(1-q_k)\int_0^{C_l} L_j(x+C_{k1})\,\mathrm{d}x \end{aligned}. \tag{17}$$

L_l can be written:

$$\begin{aligned} L_l(x) \;=\; & q_l q_k L_j(x) + q_l(1-q_k)L_j(x+C_{k1}) \\ & +(1-q_l)q_k L_j(x+C_l) + (1-q_l)(1-q_k)L_j(x+C_{k1}+C_l) \end{aligned}, \tag{18}$$

and E_{k2} is:

$$\begin{aligned} E_{k2} &= T(1-q_k)\textstyle\int_0^{C_{k2}} L_l(x)\,\mathrm{d}x \\ &= T(1-q_k)\Big\{q_l q_k \textstyle\int_0^{C_{k2}} L_j(x)\,\mathrm{d}x + q_l(1-q_k)\int_0^{C_{k2}} L_j(x+C_{k1})\,\mathrm{d}x \\ &\quad +(1-q_l)q_k\textstyle\int_0^{C_{k2}} L_j(x+C_l)\,\mathrm{d}x + (1-q_l)(1-q_k)\int_0^{C_{k2}} L_j(x+C_{k1}+C_l)\,\mathrm{d}x\Big\} \end{aligned}. \tag{19}$$

From (16), (17) and (19) it can be appreciated that, as C_{k1} increases, E_{k1} increases, and E_l and E_{k2} decrease, so that there may exist a value of C_{k1} for which $E_{k1}+E_{k2}$ and E_l from (16,17,19) match the results of (7–11) for E_k and E_l.

Instead of unit k being split by unit l, we could have considered unit l being split by unit k, which would lead to similar results. Considering the case of k being split by l, there is a maximum for E_l (corresponding to a minimum of E_k) obtained when $C_{k1}=0$. Should E_l (from (7–11) be higher than this maximum, the instance of l being split by k should be taken up.

(An important related issue is that loading a unit of capacity C and probability of failure q is not equivalent to loading two units that both have probability of failure q and capacities $C_1>0$ and $C_2>0$ such that $C_1+C_2=C$. Only if $q=0$ are both loadings equivalent.)

5.3 The multi-interval Bloom and Gallant model

As power planning for a long time period cannot take into account the 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 the variables and parameters that refer to the $i^{\underline{\text{th}}}$ interval.

Therefore, some constraints refer only to variables of a single interval, while other may refer to variables in several intervals. For example, constraints on the minimum consuption of gas may affect several or all the intervals, while emission limit constraints, or the constraint associated with the units comprising a combined-cycle unit refer to each single interval.

Overhauling of thermal units must be taken into account, so, there will be intervals where some units must be 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, and with inequality and equality non-load-matching constraints, can then be expressed as:

$$\underset{Ej^i}{\text{minimize}} \quad \sum_{i=1}^{Ni} \sum_{j=1}^{N_u} f_j E_j^i \tag{20}$$

$$\text{subject to:} \quad \sum_{j\in\omega} E_j^i \leq \widehat{E}^i - W^i(\omega) \quad \forall\omega \in \Omega^i \quad i = 1,\ldots,N_i \tag{21}$$

$$A_{\geq}^i E^i \geq R_{\geq}^i \qquad i = 1,\ldots,N_i \tag{22}$$

$$\sum_i A_{\geq}^{0i} E^i \geq R_{\geq}^0 \tag{23}$$

$$A_{=}^i E^i = R_{=}^i \qquad i = 1,\ldots,N_i \tag{24}$$

$$\sum_i A_{=}^{0i} E^i = R_{=}^0 \tag{25}$$

$$E_j^i \geq \underline{0} \qquad j = 1,\ldots,N_u, \quad i = 1,\ldots,N_i \tag{26}$$

where:

$A_{\geq}^i \in \mathbb{R}^{N_{\geq}^i \times N_u}$	matrix of coefficients of inequalities that refer only to energies of $i^{\underline{\text{th}}}$ interval,
$A_{\geq}^{0i} \in \mathbb{R}^{N_{\geq}^0 \times N_u}$	matrix of coefficients of inequalities that refer to energies of more than one interval related to energies of $i^{\underline{\text{th}}}$ interval,
$R_{\geq}^i \in \mathbb{R}^{N_{\geq}^i}$	right-hand sides of inequalities that refer only to energies of $i^{\underline{\text{th}}}$ interval,
$R_{\geq}^0 \in \mathbb{R}^{N_{\geq}^0}$	right-hand sides of inequalities that refer to energies of more than one interval
$A_{=}^i \in \mathbb{R}^{N_{=}^i \times N_u}$	matrix of coefficients of equalities that refer only to energies of $i^{\underline{\text{th}}}$ interval,
$A_{=}^{0i} \in \mathbb{R}^{N_{=}^0 \times N_u}$	matrix of coefficients of equalities that refer to energies of more than one interval related to energies of $i^{\underline{\text{th}}}$ interval,
$R_{=}^i \in \mathbb{R}^{N_{=}^i}$	right-hand sides of equalities that refer only to energies of $i^{\underline{\text{th}}}$ interval,
$R_{=}^0 \in \mathbb{R}^{N_{=}^0}$	right-hand sides 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 (23) and (25), which are the multi-interval constraints, be empty, the problem would be separable into N_i subproblems, one for each interval. Otherwise, a join solution must be found.

5.4 Approximate model of 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 variables, can be employed. In it, the whole, or a part, of the reservoir systems of one or several basins are considered as a single pseudo-thermal unit k with cost $\widetilde{f}_k=0$, outage probability $q_k=0$ and capacity C_k (normally lower than the maximum installed hydropower capacity), but having a constraint binding the intervals' hydrogenerations over the successive intervals so that they add up to a total expected hydrogeneration R_{Hk}^0 for the whole period:

$$\sum_i^{N_i} E_k^i = R_{Hk}^0 \,,$$

which is a constraint of the type (25).

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

Constraints (21) and (26) define, for each interval, a convex polyhedron whose vertices can be easily calculated. To apply the Ford-Fulkerson procedure, energies $E^i \in \mathbb{R}^{N_u}$ must be expressed as convex combinations of all vertices V_k^i of the $i^{\underline{\text{th}}}$ interval polyhedron:

$$E^i = V^i \Lambda^i, \quad V^i \in \mathbb{R}^{N_u \times N_V^i} \qquad \Lambda^i \geq \underline{0}, \qquad \mathbb{I}' \Lambda^i = 1 \quad \forall i$$

$\mathbb{I}' = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$ being the all one vector.

The number N_V^i of vertices of one such polyhedron is very high as the number of constraints (21) that define it, jointly with the nonnegativity bounds (26), 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.

Problem (20-26) can be rewritten as:

$$\underset{\Lambda^i}{\text{minimize}} \quad \sum_{i=1}^{N_i} f' V^i \Lambda^i \tag{27}$$

$$\text{subject to:} \quad \mathbb{I}' \Lambda^i = 1 \quad i = 1, \dots, N^i \tag{28}$$

$$\left. \begin{array}{l} A_=^i V^i \Lambda^i = R_=^i \\ A_\geq^i V^i \Lambda^i \geq R_\geq^i \end{array} \right\} \quad i = 1, \dots, N^i \tag{29}$$

$$\sum_{i=1}^{Ni} A_=^{0\,i} V^i \Lambda^i = R_=^0 \tag{30}$$

$$\sum_{i=1}^{Ni} A_\geq^{0\,i} V^i \Lambda^i \geq R_\geq^0 \tag{31}$$

$$\Lambda^i \geq \underline{0} \quad i = 1, \dots, N^i \tag{32}$$

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

The convex coefficients $\Lambda^i \in \mathbb{R}^{N_V^i}$, i=1, …, N_i are only a part of the problem variables. The rest of the variables are the surpluses $S^i \in \mathbb{R}^{N_\geq^i}$, i=1, …, N_i of the inequalities that refer to energies of a single interval, and the surpluses $S^0 \in \mathbb{R}^{N_\geq^0}$ of the inequalities that refer to energies of several intervals.

The problem to be solved by the FFcg method is thus:

$$\underset{S^0,\, S^i,\, \Lambda^i}{\text{minimize}} \quad \sum_{i=1}^{N_i} f^{i\,\prime} V^i \Lambda^i \tag{33}$$

$$\text{subject to:} \quad \mathbb{I}' \Lambda^i = 1 \quad i = 1, \dots, N^i \tag{34}$$

$$\left. \begin{array}{l} A_=^i V^i \Lambda^i = R_=^i \\ A_\geq^i V^i \Lambda^i - S^i = R_\geq^i \end{array} \right\} \quad i = 1, \dots, N^i \tag{35}$$

$$\sum_{i=1}^{Ni} A_=^{0\,i} V^i \Lambda^i = R_=^0 \tag{36}$$

$$\sum_{i=1}^{Ni} A_\geq^{0\,i} V^i \Lambda^i - S^0 = R_\geq^0 \tag{37}$$

$$S^0 \geq \underline{0} \qquad S^i \geq \underline{0}, \quad \Lambda^i \geq \underline{0} \quad i = 1, \dots, N^i\,. \tag{38}$$

6.1 Application of Ford-Fulkerson's procedure

Since the problem is linear with linear equalities and nonnegativity bounds on all variables, the active constraint matrix $\widehat{A}$ at a basic feasible point can be reordered so that the basic and nonbasic columns and variables are separated:

$$\widehat{A}X = \begin{bmatrix} B & N \\ \mathbb{0} & \mathbb{1} \end{bmatrix} \begin{bmatrix} X_B \\ X_N \end{bmatrix} = \begin{bmatrix} R \\ \underline{0} \end{bmatrix} \tag{39}$$

Subindices $_B$ and $_N$ will be used in order to tell the basic variables from the nonbasic variables. The number of basic variables is that of constraints: $N_i{+}N_={+}N_\geq$ (and is independent of the number of vertices!): $B{\in}\mathrm{I\!R}^{(N_i+N_=+N_\geq)\times(N_i+N_=+N_\geq)}$. The number of nonbasic variables is the total number of variables minus the basic ones: $\sum_i N_V^i{+}N_\geq{-}N_i{-}N_={-}N_\geq{=}\sum_i N_V^i{-}N_i{-}N_=$ which is a function of the number of vertices and will be extrenely large even for a modest N_u.

Let Λ_B^i be the basic convex coefficients and S_B^i and S_B^0 the basic surpluses of the inequality constraints. The basic matrix B and the basic variables will satisfy:

$$\begin{bmatrix}
\mathbb{I}' & & & & & & & \\
 & & \mathbb{I}' & & & & & \\
 & & & & \cdots & & & \\
 & & & & & \mathbb{I}' & & \\
A_=^1 V_B^1 & & & & & & & \\
A_\geq^1 V_B^1 & -\mathbb{1}_B^1 & & & & & & \\
 & & A_=^2 V_B^2 & & & & & \\
 & & A_\geq^2 V_B^2 & -\mathbb{1}_B^2 & & & & \\
 & & & & \cdots & & & \\
 & & & & & A_=^{N_i} V_B^{N_i} & & \\
 & & & & & A_\geq^{N_i} V_B^{N_i} & -\mathbb{1}_B^{N_i} & \\
A_=^{0\,1} V_B^1 & & A_=^{0\,2} V_B^2 & & \cdots & A_=^{0\,N_i} V_B^{N_i} & & \\
A_\geq^{0\,1} V_B^1 & & A_\geq^{0\,2} V_B^2 & & \cdots & A_\geq^{0\,N_i} V_B^{N_i} & & -\mathbb{1}_B^0
\end{bmatrix}
\begin{bmatrix} \Lambda_B^1 \\ S_B^1 \\ \Lambda_B^2 \\ S_B^2 \\ \vdots \\ \Lambda_B^{N_i} \\ S_B^{N_i} \\ S_B^0 \end{bmatrix}
=
\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ R_=^1 \\ R_>^1 \\ R_=^2 \\ R_\geq^2 \\ \vdots \\ R_=^{N_i} \\ R_>^{N_i} \\ R_=^0 \\ R_\geq^0 \end{bmatrix} \tag{40}$$

where the matrix on the left is B and the *rhs* vector contains the independent terms of the constraints.

The nonbasic submatrix of the constraints is:

$$N = \begin{bmatrix}
\mathbb{I}' & & & & & & & \\
 & & \mathbb{I}' & & & & & \\
 & & & & \cdots & & & \\
 & & & & & \mathbb{I}' & & \\
A_=^1 V_N^1 & & & & & & & \\
A_\geq^1 V_N^1 & -\mathbb{1}_N^1 & & & & & & \\
 & & A_=^2 V_N^2 & & & & & \\
 & & A_\geq^2 V_N^2 & -\mathbb{1}_N^2 & & & & \\
 & & & & \cdots & & & \\
 & & & & & A_=^{N_i} V_N^{N_i} & & \\
 & & & & & A_\geq^{N_i} V_N^{N_i} & -\mathbb{1}_N^{N_i} & \\
A_=^{0\,1} V_N^1 & & A_=^{0\,2} V_N^2 & & \cdots & A_=^{0\,N_i} V_N^{N_i} & & \\
A_\geq^{0\,1} V_N^1 & & A_\geq^{0\,2} V_N^2 & & \cdots & A_\geq^{0\,N_i} V_N^{N_i} & & -\mathbb{1}_N^0
\end{bmatrix} \tag{41}$$

and the vector of nonbasic variables is:

$$\Lambda_N^1 \; S_N^1 \; \Lambda_N^2 \; S_N^2 \; \ldots \; \Lambda_N^{N_i} \; S_N^{N_i} \; S_N^0$$

Here follows an example of active set $\widehat{A}$ with $N_i{=}3$, $N_=^1{=}1$, $N_\geq^1{=}2$ and both basic surpluses s_k^1 being basic, $N_=^2{=}2$, $N_\geq^1{=}2$ one of the surpluses s_k^2 being basic and the other nonbasic, $N_=^3{=}1$, $N_\geq^3{=}1$

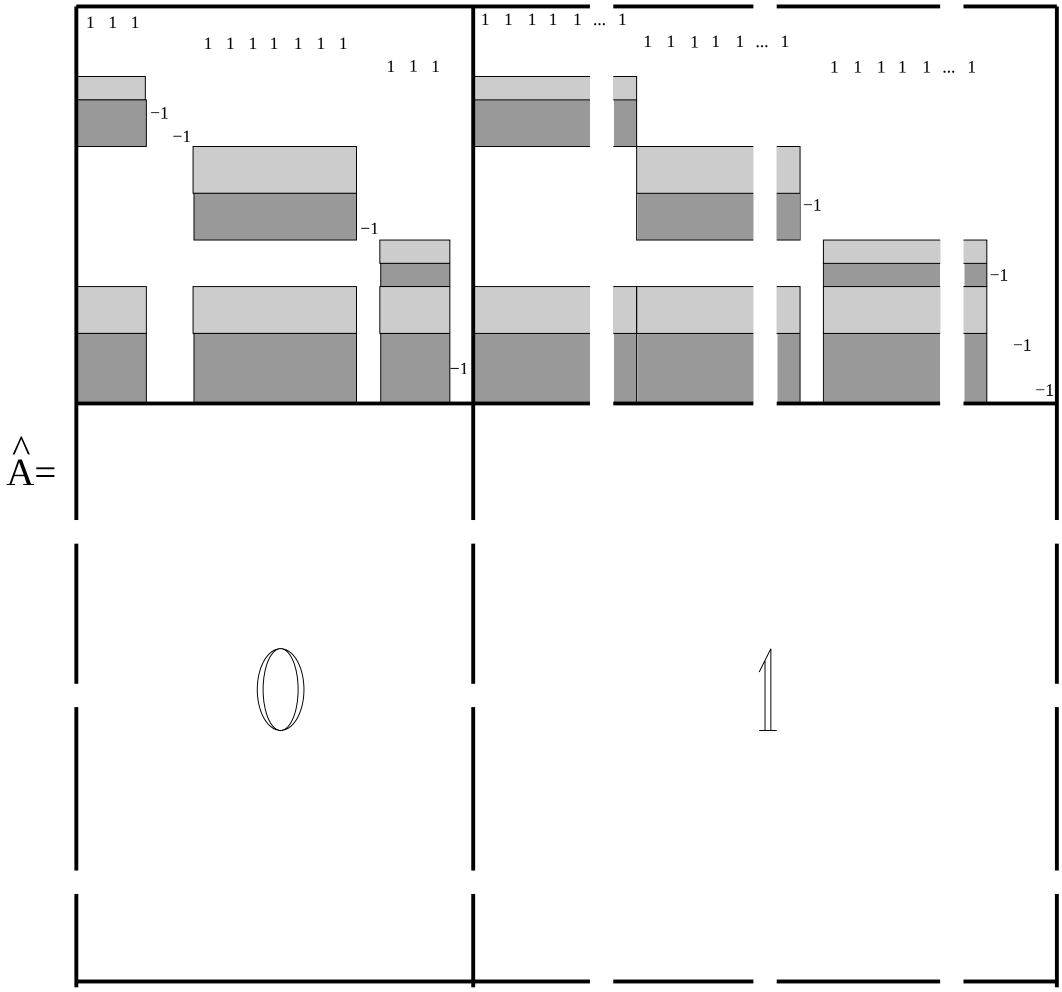

Figure 4: Example of the active constraint matrix for a case with N_i=3 and $N_=+N_\geq$=14.

with nonbasic surplus, and $N^0_=$=2 and $N^0_\geq$=3 with one surplus s^0_k being basic and the other two nonbasic; see Fig. 4. There are 17 basic variables in all (as there are 17 constraints, 3=N_i of these are the coefficient convexity constraints of λ^i_j). Of the 17 basic variables, there are 13 convex coefficients λ^i_j of vertices and there are 4 surpluses. These numbers of coefficients and surpluses may change with iterations, but they will always add up to 17.

6.2 Calculation of Lagrange multipliers

In problems with equality constraints and nonnegativity bounds they are calculated as follows:

$$\begin{bmatrix} B' & \mathbb{0} \\ N' & \mathbb{1} \end{bmatrix} \begin{bmatrix} \Pi \\ \Sigma \end{bmatrix} = \begin{bmatrix} C_B \\ C_N \end{bmatrix} \quad \rightarrow \quad \left\{ \begin{array}{c} B'\Pi = C_B \ \rightarrow \ \Pi \\ \Sigma = C_N - N'\Pi \end{array} \right. \tag{42}$$

From the first equation we have:

$$
\begin{bmatrix}
\mathbb{I} & & V_B^{1\prime} A_=^{1\prime} & V_B^{1\prime} A_\geq^{1\prime} - \mathbb{1}_B^{1\,\tau} & & & & V_B^{1\prime} A_=^{0\,1\prime} & V_B^{1\prime} A_\geq^{0\,1\prime} \\
\vdots & & & & \vdots & & & & \\
& \mathbb{I} & & & & V_B^{N_i\prime} A_=^{N_i\prime} & V_B^{N_i\prime} A_\geq^{N_i\prime} - \mathbb{1}_B^{N_i\,\tau} & V_B^{N_i\prime} A_=^{0\,N_i\prime} & V_B^{N_i\prime} A_\geq^{0\,N_i\prime} \\
& & & & & & & & -\mathbb{1}_B^{0\,\prime}
\end{bmatrix}
\times
\begin{bmatrix}
\pi_\lambda^1 \\ \cdots \\ \pi_\lambda^{N_i} \\ \Pi_=^1 \\ \Pi_\geq^1 \\ \cdots \\ \Pi_=^{N_i} \\ \Pi_\geq^{N_i} \\ \Pi_=^0 \\ \Pi_\geq^0
\end{bmatrix}
=
\begin{bmatrix}
V_B^{1\prime} f^1 \\ \underline{0} \\ \vdots \\ V_B^{N_i\prime} f^{N_i} \\ \underline{0} \\ \underline{0}
\end{bmatrix}
\tag{43}
$$

From this equation we obtain the expected result that all components of multipliers $\Pi_\geq^i$ $i{=}1,2,\ldots,N_i,0$ corresponding to nonactive inequality constraints are zero.

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

$$
\begin{gathered}
\pi_\lambda^i + V_{N\,k}^i{}'(A_=^i{}'\Pi_=^i + A_\geq^i{}'\Pi_\geq^i + A_=^{0\,i\prime}\Pi_=^0 + A_\geq^{0\,i\prime}\Pi_\geq^0) + \sigma_{V\,k}^i = V_{N\,k}^i{}' f^i \\
i = 1,2,\ldots,N_i
\end{gathered}
$$

hence:

$$
\begin{gathered}
\sigma_{V\,k}^i = (f^i - A_=^i{}'\Pi_=^i - A_\geq^i{}'\Pi_\geq^i - A_=^{0\,i\prime}\Pi_=^0 - A_\geq^{0\,i\prime}\Pi_\geq^0)'V_{N\,k}^i - \pi_\lambda^i \\
i = 1,2,\ldots,N_i
\end{gathered}
\tag{44}
$$

i.e., there is a nonbasic vertex of interval i if $\sigma_{V\,k}^i{<}0$, and this will be so if for the modified costs $f^i - A_=^i{}'\Pi_=^i - A_\geq^i{}'\Pi_\geq^i - A_=^{0\,i\prime}\Pi_=^0 - A_\geq^{0\,i\prime}\Pi_\geq^0$ the vector of energies $E_k^i{=}V_{N\,k}^i$ yields a cost lower than π_λ^i.

The other equation we obtain from the Σ equations is:

$$
-\pi_{l\geq}^i + \sigma_{S\,l}^i = 0 \;\rightarrow\; \sigma_{S\,l}^i = \pi_{l\geq}^i \quad i = 1,2,\ldots,N_i,0
$$

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

6.3 Efficiency of the obtention of new vertices

The problem of finding a (nonbasic) vertex E_k^i such that

$$
\{f^i - A_=^i{}'\Pi_=^i - A_\geq^i{}'\Pi_\geq^i - A_=^{0\,i\prime}\Pi_=^0 - A_\geq^{0\,i\prime}\Pi_\geq^0\}'E_k^i < \pi_\lambda^i \quad i = 1,2,\ldots,N_i
$$

appears to be simple, because given

$$
\widehat{f^i} = f^i - A_=^i{}'\Pi_=^i - A_\geq^i{}'\Pi_\geq^i - A_=^{0\,i\prime}\Pi_=^0 - A_\geq^{0\,i\prime}\Pi_\geq^0 \tag{45}
$$

it is straightforward to sort the elements of $\widehat{f^i}$ in increasing value determining a loading order, and compute the elements of E_k^i by successive convolution (1) and integration (2). (And checking whether $\widehat{f^i}{}' E_k^i{<}\pi_\lambda^i$ 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.

6.4 Creation of vertices with some zero energy elements

There are vertices that cannot be generated by the former procedure, as all components of E^i_k will be within its bounds (6) and there may be vertices required that have a given component of zero or, in general, lower than the lower bound in (6). This is the case with some intervals when there are fuel or generation upper limits over several intervals for a given unit, and the optimizer is such that this unit generates zero at some of the intervals.

It is thus necessary to create vertices with zero generation units at certain intervals as if these were not available (even though they have no programmed overhauling in the interval). Otherwise, the optimizer will not be reached.

The procedure for generating vertices with zero generation elements is simple. Once the modified costs $\widehat{f^i}$ (45) have been obtained and sorted in increasing modified cost order, and the energies E^i_k have been calculated using this order, the total modified cost $\sum_{j=1}^{N^i_u} \widehat{f^i_j} E^i_{k\,j}$ is computed and checked with π^i_λ (44) to find out whether

$$\sigma^i_{V\,k} = \sum_{j=1}^{N^i_u} \widehat{f^i_j} E^i_{k\,j} - \pi^i_\lambda < 0$$

If this is not the case, a vertex without any number of the last loaded units (in increasing modified cost order), as if these were not available, would be equally acceptable. Given that any component $E^i_{k\,j}{>}0$, and that the last terms of $\widehat{f^i}$ (in increasing modified cost order) might be positive, excluding these terms from the sumation (by making zero the corresponding component $E^i_{k\,j}$) will help to obtain a negative $\sigma^i_{V\,k}$.Excluding terms with $\widehat{f^i_j}{<}0$ would act against this goal.

The following algorithm could thus be employed to obtain acceptable vertices with zero components where appropriate.

Using the variables:

`negsig`	variable employed to compute a negative $\sigma^i_{V\,k}$ for a vertex E^i_k of the $i^{\underline{\text{th}}}$ interval
`ior(·)`	the order vector of increasing modified costs $\widehat{f^i}$
`everti(·)`	unit energies of a vertex E^i_k of the $i^{\underline{\text{th}}}$ interval (initially containing the vertex calculated with no zero elements, on end it may contain some zero elements, if necessary)
`modcost(·)`	vector of modified costs $\widehat{f^i}$
`neg_to_pos`	logical variable that indicates when `negsig` would change from negative to positive
`Nui`	number of available units in interval i (dimension of vectors `everti`, `ior` and `modcost`)
`pii`	π^i_λ

the following code will obtain the correct vertex, if any, `everti(·)`:

```
neg_to_pos=.false.
negsig=-pii
for k=1 until Nui do
  jor=ior(k)
  aux=modcost(jor)*everti(jor)
  if (negsig<0 and aux>-negsig) neg_to_pos=.true.
  if (neg_to_pos=.false.)  negsig=negsig+aux
  if (aux>0 and neg_to_pos=.true.)  everti(jor)=0
endfor
if (negsig<0) then
  exit !everti(·) is an acceptable vertex
else
  exit !a vertex of a different interval must be tried
endif
```

If no acceptable vertex can be found for any interval and no nonbasic surplus variable should increase its value, the current point is the optimizer.

6.5 Ford-Fulkerson column generation algorithm

Although Ford and Fulkerson designed their procedure for multicommodity network-flow problems [6], it can be generalized to other linearly constrained problems.

For a linear objective function problem with N_i stages formulated as (33-38) the FFcg algorithm starting from a basic feasible point would be:

0) Let $\{\Lambda^i_B, S^i_B, S^0_B,\ i= 1,\dots,N_i\}$ be the set of basic variables, and let B be the basic matrix, whose structure is as that in (40)

1) Solve system (43) to compute the Lagrange multipliers π^i_λ of the convexity equations (34), and $\Pi^i_=, \Pi^i_\geq\ , i{=}1,\dots,N_i$ and $\Pi^0_=\ \Pi^0_\geq$ of constraints (35-38)

2) Pricing of nonbasic surplusses:
   - if $\Pi^i_\geq<\underline{0}\ , i{=}1,\dots,N_i$ or $\Pi^0_\geq<\underline{0}\ \rightarrow\ \exists\, s^i_l$ for some $i{=}1,\dots,N_i,0$ that should enter the basis; go to step 4)
   - ◦ if $\Pi^i_\geq\geq\underline{0}\ , i{=}1,\dots,N_i$ and $\Pi^0_\geq\geq\underline{0}$ (no nonbasic surplus can enter the basis); go to step 4)

3) Pricing of new nonbasic vertices for the intervals $i{=}1,\dots,N_i$:
   - 3.1) form the modified cost vector $\widehat{f^i}$ (45)
   - 3.2) sort the elements of $\widehat{f^i}$ and determine energies E^i_k of new nonbasic vertex k through convolution (1) and integration (2)
   - if for some i $\sigma^i_{V\,k}{=}\widehat{f^i}'E^i_k{-}\pi^i_\lambda{<}0$ vertex E^i_k of interval i should enter the basis; go to step 5)
   - ◦ if for all i $\sigma^i_{V\,k}{=}\widehat{f^i}'E^i_k{-}\pi^i_\lambda{\geq}0$ (no vertex of any interval can enter the basis); **[END]**: the current basic set is optimal

4) Compute change of basic variables when a surplus enters the basis
   - 4.1) being $\underline{0}{\in}\mathbb{R}^{N_i}$ and $\mathcal{E}_l$ the $l^{\underline{\text{th}}}$ column of $\mathbb{1}{\in}\mathbb{R}^{N_=+N_\geq}$, solve for the direction of change

$$B\begin{bmatrix}\Delta\Lambda_B\\ \Delta S_B\end{bmatrix}=\begin{bmatrix}\underline{0}\\ \mathcal{E}_l\end{bmatrix}$$

   - 4.2) compute maximum step length $\overline{\alpha}$ for basic convex coefficient and basic surplus decrease

$$\overline{\alpha}=\min\left\{\frac{\lambda^i_{B\,j}}{\Delta\lambda^i_{B\,j}}\,\forall\, j\ i=1,\dots,N_i\,\middle|\,\Delta\lambda^i_{B\,j}<0\ ,\ \frac{s^i_{B\,k}}{\Delta s^i_{B\,k}}\,\forall\, k\ i=1,\dots,N_i,0\,\middle|\,\Delta s^i_{B\,k}<0\right\}$$

   - ⋆ if $\overline{\alpha}{=}\frac{\lambda^i_{B\,j}}{\Delta\lambda^i_{B\,j}}$ vertex j of $i^{\underline{\text{th}}}$ interval leaves the basis
   - ⋆ if $\overline{\alpha}{=}\frac{s^i_{B\,k}}{\Delta s^i_{B\,k}}$ surplus $s^i_{B\,k}$ leaves the basis
   - 4.3) set the new basic surplus value $s^i_{B\,l}{=}\overline{\alpha}$ and go to 6)

5) Compute change of basic variables when a vertex enters the basis
   - 5.1) being $\mathcal{E}_i$ the $i^{\underline{\text{th}}}$ column of $\mathbb{1}{\in}\mathbb{R}^{N_i}$ and $V{\in}\mathbb{R}^{N_=+N_\geq}$ the product of all constraint coefficients of (22-26) by the new vertex E^i_k, solve for the direction of change

$$B\begin{bmatrix}\Delta\Lambda_B\\ \Delta S_B\end{bmatrix}=-\begin{bmatrix}\mathcal{E}_i\\ V\end{bmatrix}$$

5.2) compute maximum step length for basic convex coefficient and basic surplus decrease $\overline{\alpha}$

$$\overline{\alpha} = \min\left\{ \frac{\lambda^i_{Bj}}{\Delta\lambda^i_{Bj}} \forall j \; i=1,\ldots,N_i \,\middle|\, \Delta\lambda^i_{Bj} < 0 \;,\; \frac{s^i_{Bk}}{\Delta s^i_{Bk}} \forall k \; i=1,\ldots,N_i,0 \,\middle|\, \Delta s^i_{Bk} < 0 \right\}$$

- ⋆ if $\overline{\alpha}=\frac{\lambda^i_{Bj}}{\Delta\lambda^i_{Bj}}$ vertex j of i^{th} interval leaves the basis
- ⋆ if $\overline{\alpha}=\frac{s^i_{Bk}}{\Delta s^i_{Bk}}$ surplus s^i_{Bk} leaves the basis

5.3) set the convex coefficient value $\lambda^i_{Bk}=\overline{\alpha}$ of the new basic vertex and go to 6)

6) Update old basic variables:

$$\begin{aligned} \Lambda_B &:= \Lambda_B + \overline{\alpha}\Delta\Lambda_B \\ S_B &:= S_B + \overline{\alpha}\Delta S_B \end{aligned}$$

7) Update basic set, discard leaving basic column in B, substitute it by new basic column, and return to 1)

It should be noted that the only place in the algorithm where operations are specific to the problem solved is step 3.2, which is the obtention of the values that characterize a new vertex in the long-term electric planning problem. The rest of the steps are the standard FFcg procedure.

6.6 Differences between Ford-Fulkerson's and Dantzig-Wolfe's column generation algorithms

The two methods are similar, and Dantzig and Wolfe admit in their article [5] that the Ford and Fulkerson proposal [6] *inspired* their work. The main difference with the Dantzig and Wolfe method is that the discarded basic columns are kept as nonbasic columns $\widehat{N}$ and, before generating a new one, existing nonbasic columns are priced to check if one of them could enter the basis. This step avoids computing the same column more than once, which, in our problem, requires finding a new vertex through sorting modified costs, convolution and integration, and multiplying constraints (22-25) by the vertex.

To adapt the Ford-Fulkerson algorithm introduced in the section above to the Dantzig-Wolfe procedure, some small changes should be introduced. A new step between 3.1 and 3.2 must be added, and step 3) must be about the pricing of old and new nonbasic vertices:

3) Pricing of old and new nonbasic vertices for the intervals i=1, …, N_i:

3.1) form the modified cost vector $\widehat{f^i}$ (45)

3.1') price old nonbasic vertices $\widehat{E}^i_j$: $\widehat{\sigma}^i_{Vj}=\widehat{f^i}'E^i_j-\pi^i_\lambda$

- if for some i $\widehat{\sigma}^i_{Vj}<0$ vertex $\widehat{E}^i_j$ of interval i should enter the basis; go to step 5)
- ∘ if for i $\widehat{\sigma}^i_{Vj}\geq 0$ (no old vertex of interval i can enter the basis) go to 3.2)

3.2) sort the elements of $\widehat{f^i}$ and determine energies E^i_k of new nonbasic vertex k through convolution (1) and integration (2)

- if for some i $\sigma^i_{Vk}=\widehat{f^i}'E^i_k-\pi^i_\lambda<0$ vertex E^i_k of interval i should enter the basis; go to step 5)
- ∘ if for all i $\sigma^i_{Vk}=\widehat{f^i}'E^i_k-\pi^i_\lambda\geq 0$ (no vertex of any interval can enter the basis); $\boxed{\text{END}}$: the current basic set is optimal

Step 5.1) should now read:

5.1) being $\mathcal{E}_i$ the i^{th} column of $1\!\!1\in\mathbb{R}^{N_i}$ and $V\in\mathbb{R}^{N_=+N_\geq}$ the product of all constraint coefficients of (22-26) by the old ($\widehat{E}^i_j$) or new vertex (E^i_k) as appropriate, solve for the direction of change

Also $\widehat{N}$ must be updated, so step 7 should be:

7) Update basic set, discard leaving basic column in B, substitute it by new basic column. If necessary, update nonbasic set $\widehat{N}$. Return to 1).

6.7 Obtention of a 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 an active surplus for the constraint added or a new vertex, until all constraints are satisfied. The details of this process can be found in [11]. (Inequality constraints are here considered to be $\geq$.)

A point satisfying all load-matching constraints for a given interval (a vertex) is easily obtained by calculating, through convolution and integration, the energies of the available units loaded in any order. The initial basis of the FFcg procedure is thus $B{=}\mathbb{1}^{N_i \times N_i}$, where each 1 in B is the convex coefficient of the only vertex of each interval. Fig. 5 shows the initial basis B and the initial matrix of basic vertices V_B.

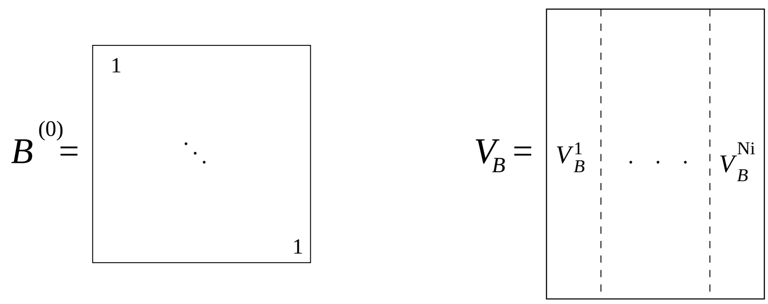

Figure 5: Initial basic matrix

Let us assume that we have a former basis and that a new inequality constraint must be satisfied. If, with the current basis and feasible point, the constraint is already satisfied, a surplus becomes basic. Otherwise a non-basic vertex must be found. At the end of the feasibility process the basic matrix dimension shall be $B \in \mathbb{R}^{(N_i+N_=+N_\geq)\times(N_i+N_=+N_\geq)}$. (A satisfied equality constraint could also be incorporated into the base by adding a degenerate zero surplus column.)

Suppose that a basic matrix $B^{(g-1)}$ for the first $g{-}1$ constraints is available and it has to be extended for the $g^{\underline{\text{th}}}$ constraint, with coefficients A^g, which is some non-load-matching constraint (22-25) with nonzero coefficients in one or more intervals. A new basic variable must be determined to extend the basis as depicted in Fig. 6

where $C_N = [\mathbb{1}_i \,|\, V'_{Nk} A^{g\prime} \,|\, 0]'$, $\mathbb{1}_i$ being the $i^{\underline{\text{th}}}$ row of $\mathbb{1}{\in}\mathbb{R}^{N_i \times N_i}$, when the new basic variable is a vertex from the $i^{\underline{\text{th}}}$ interval, or the new basic variable is simply the surplus of the $l^{\underline{\text{th}}}$ constraint added, in which case $C_N = [\underline{0}\,| - \mathbb{1}'_l]'$; l could correspond to the constraint being added g.

In general, one of the following cases will take place when the available basis $B^{(g-1)}$, which makes feasible the first $g{-}1$ constraints, is extended to include the $g^{\underline{\text{th}}}$ constraint with coefficients A^g:

- the $g^{\underline{\text{th}}}$ constraint is satisfied with the current basis
  - the $g^{\underline{\text{th}}}$ constraint is an equality
    The (degenerate) $g^{\underline{\text{th}}}$ surplus is the new basic variable (with zero value, and will be eliminated whenever it becomes non-basic).
  - the $g^{\underline{\text{th}}}$ constraint is an inequality
    The new basic variable is the surplus of the $g^{\underline{\text{th}}}$ constraint.

  In both cases, the column added to the basic matrix is $C_N{=}[0 \cdots 0 - 1]'$.

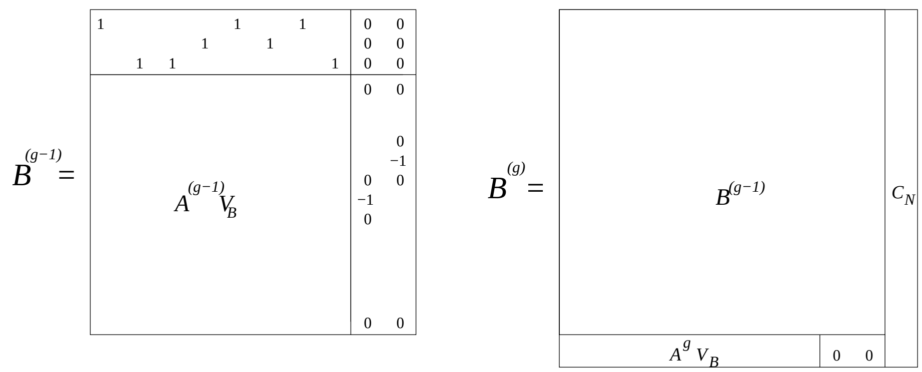

Figure 6: Scheme of the basic matrix expansion for a case with three intervals, having 8 basic vertices and 2 basic surpluses

- the $g^{\underline{\text{th}}}$ constraint is not satisfied with the current basis
  There are also two cases and, in both, the goal is satisfying the new constraint as an equality. Let us assume that $A^g V \Lambda < r^g$ when the $g^{\underline{\text{th}}}$ constraint is an equality or a $\geq$ constraint. Then we must solve the problem

$$\begin{aligned} \underset{\Lambda}{\text{minimize}} \quad & -A^g V \Lambda \\ \text{subject to:} \quad & \begin{bmatrix} A_{=}^{(g-1)} V & \mathbb{0} \\ A_{\geq}^{(g-1)} V & -\mathbb{1} \end{bmatrix} \begin{bmatrix} \Lambda \\ S \end{bmatrix} = \begin{bmatrix} R_{=}^{(g-1)} \\ R_{\geq}^{(g-1)} \end{bmatrix} \\ & \mathbb{I} \Lambda^i = 1 \qquad i = 1, \dots, N_i \\ & \Lambda, S \geq 0 \end{aligned} \tag{46}$$

  Should $A^g V_B \Lambda > r^g$ and g be an equality or $\leq$ constraint, the opposite must be solved. The problem is the same except the objective function (46) which is now

$$\underset{\Lambda}{\text{minimize}}\, A^g V \Lambda$$

  A single implementation is possible for both cases and only special attention must be paid to the sign criterion of the Lagrange multipliers calculated. This problem is solved by the Simplex algorithm and is stopped when a base satisfying $A^g V = r^g$ is achieved. The steps are:

  - Look for a candidate non-basic variable to become basic (through the Lagrange multiplier signs)
  - Compute the increase/decrease of basic variables due to the increase of the entering variable
  - Stop before the maximum step is reached if the entering constraint g is satisfied as an equality, upon which the process is stopped having added the $g^{\underline{\text{th}}}$ constraint and the nonbasic variable column in the basis.

  The calculation of Lagrange multipliers is as explained in sections 6.2 and 6.3. The candidate variable could be a vertex V_{Nk}^i of some interval i in which the $g^{\underline{\text{th}}}$ constraint has some element, or a surplus s_l of the $l^{\underline{\text{th}}}$ active constraint. When some variable changes its value the others modify their value proportionally. Once the new candidate variable is known, the changes in the basic variables must be calculated as:

$$B^{(g-1)} \begin{bmatrix} \Delta \Lambda_B \\ \Delta S_B \end{bmatrix} = -C_N$$

where C_N is the column associated with the new basic variable in the basis, $B^{(g)}$:

$$C_N = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow (N_i + l) \text{ position, if } \sigma^i_{Sl} < 0 \quad \text{or}$$

$$C_N = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ A^g V^i_{Nk} \end{bmatrix} \leftarrow i^{\underline{\text{th}}} \text{ position, if } \sigma^i_{Vk} < 0$$

The largest step, $\overline{\alpha}$, that basic variables can take without being out of bounds is,

$$\overline{\alpha} = \min\left\{ -\frac{\lambda_k}{\lambda_k} \,\middle|\, \forall \Delta\lambda_k < 0, -\frac{s_l}{\Delta s_l} \,\middle|\, \forall \Delta s_l < 0 \right\}$$

Moreover, as we wish to satisfy $A^g V_B \Lambda = r^g$, the maximum increase that the new basic variable should achieve must be calculated as

$$\alpha_{new} = \frac{r^g - A^g V_B \Lambda_B}{A^g (V_B \Delta\Lambda_B + V^i_{Nk})}$$

if V^i_{Nk} is the new candidate vertex.

Should the new basic variable be the surplus , we would have:

$$\alpha_{new} = \frac{r^g - A^g V_B \Lambda_B}{A^g V_B \Delta\Lambda_B}.$$

If $\alpha_{new} \leq \overline{\alpha}$, a new non-basic variable, V^i_{Nk} or s_l has been found that satisfies the $g^{\underline{\text{th}}}$ constraint as an equality. In case $\overline{\alpha} < \alpha_{new}$, the candidate variable replaces a basic variable and another iteration must be performed.

The updating of variables is,

$$\begin{cases} \lambda_j = \lambda_j + \alpha\Delta\lambda_j & , j \in B \\ s_j = s_j + \alpha\Delta s_j & , j \in B \\ s_l = \alpha & , \text{ if the candidate variable is a surplus, or} \\ \lambda^i_k = \alpha & , \text{ if the candidate variable is a vertex,} \end{cases}$$

where the step length is $\alpha = \min\{\alpha_{new}, \overline{\alpha}\}$.

Obtaining a feasible point is similar to minimizing costs with a different objective function. Therefore, the procedure for creating vertices with zero or very small generation described in 6.4 must also be employed. If when one is trying to get feasibility with respect to a given constraint, no vertex with negative $\sigma^i_{V\,k}$ can be obtained for all intervals, and no nonbasic surplus should increase its value, the problem is infeasible.

7 Computational results

7.1 Test cases

The characteristics of the test cases employed are summarized in Table 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.

Table 1: Test cases for long-term electric power planning

case	N_i	N_u	$\sum_i N_u^i$	$\sum_i N_=^i$	$N_=^0$	$\sum_i N_\geq^i$	$N_\geq^0$	$\sum_i (2^{N_u^i} - 1)$	$\sum_i \sum_j^{N_u+1} f_j E_j^i$ solver	(€)
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
ltp08	14	40	543	42	0	141	30	10599979286510	ac.set	3118958218

Cases ltp01 to ltp04, with 13, 15, 17 and 18 units, are the same problem with more or less aggregation of units and there are two subcases of each case with a different number of constraints. They were employed in another report by the authors [10] to test an *AMPL* [7] plus *Cplex 7.5* [4] direct linear programming solution of these problems, and the objective function value there obtained will be checked with that obtained using the FFcg and active set procedures. Extra details about these problems can be found in the reference given.

7.2 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 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 1. The last three columns in Table 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 (21).

Several conclusions can be drawn from the results of Table 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 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 except ltp08). Four agreement digits would be fairly acceptable from

Table 2: Comparison of the active set, and the Ford-Fulkerson column generation method

	active set method				Ford-Fulkerson column generation						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.
ltp08	4311	5593	43724	–	726	6296	629.	2025	112	9	n.a.	n.a.	n.a.

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 (except in case ltp08).

Through the `-pg` option in the Fortran compilation of the programs and the standard Unix program `gprof` (profiling), it is possible to analyze where the CPU time is spent during execution. It was found that most of the execution time of the active set implementation (over 90%) went to calculating the *rhs*'s of the new active constraints tried and, on average, about 20 new constraints are tried per iteration. With the FFcg implementation almost as much computation (about 80%) is due to calculating new vertices, which involve the same routines of convolution and integration as the calculation of the *rhs*'s. However in the FFcg the number of vertices generated per iteration is less than one, as in many iterations a slack variable is made active, and the number of iterations required has been always below that of the active set procedure.

It is not surprising that case ltp06, though bigger than cases ltp02 and ltp03, and requiring more iterations than former cases, takes less time to convergence. This is because the convolutions are much shorter in ltp06 than in the other cases because the load to be matched (of a single company in Germany) is much lower than that of the Spanish power pool, and a uniform 1 MW step is taken for storing the probability distributions of load still to be supplied, and for integration.

7.3 Performance when vertices with very small or zero energy elements were not obtained

Only in one of the test cases reported was a vertex with a forced zero generation created. It wass in case ltp01a. Should the conditionals in the loop of the algorithm of section 6.4 be commented out, the result obtained had the same objective function value and the same number of basic vertices, but it was achieved after a much bigger number of iterations (144) and having generated many more vertices (314).

Using the type of algorithm presenyted is a necessary precaution.

8 Conclusions

- The problem of long-term hydrothermal planning of electricity generation has been presented and an extension of the Bloom and Gallant model has been put forward to solve it.
- The FFcg procedure applied to the Bloom and Gallant multi-interval long-term power planning has been presented, including:
  - the general procedure
  - a new algorithm for the creation, whenever necessary, of vertices with some zero generating units
  - the procedure to find an initial feasible point.
- The computational experience with:
  - The comparison of performance between the FFcg, the active set procedure and the solution with Cplex 7.5 (when available), which shows that the FFcg procedure is the most efficient
  - The analysis of the compared efficiency of the active set and the FFcg procedures when applied to the Bloom and Gallant formulation of the multi-interval long-term, power planning
  - The comparison of precision, which shows that the FFcg reaches a precision similar to that achieved with the active set method.
  - An analysis of the influence of the creation of vertices with zero energy components.
- The FFcg procedure appears to be capable of solving real size problems (with $N_u \approx 150$ and $N_i \approx 30$) in reasonable time.

9 Acknowledgements

The authors wish to thank prof. J. Gondzio of Univ. of Edinburgh, Edinburgh G.B. for pointing out to them the differences between the Ford-Fulkerson and the Dantzig-Wolfe column generation procedures.

10 Glossary of symbols

$A^i_{\geq}$	matrix of coefficients of inequalities that refer only to energies of $i^{\underline{\text{th}}}$ interval
$A^{0i}_{\geq}$	matrix of coefficients of inequalities that refer to energies of more than one interval related to energies of $i^{\underline{\text{th}}}$ interval
$A^i_{=}$	matrix of coefficients of equalities that refer only to energies of $i^{\underline{\text{th}}}$ interval
$A^{0i}_{=}$	matrix of coefficients of equalities that refer to energies of more than one interval related to energies of $i^{\underline{\text{th}}}$ interval
B	(subindex) indication of basic variable
C_j	power capacity in MW of $j^{\underline{\text{th}}}$ generating unit
E^i_j	energy generated in MWh by unit j over the $i^{\underline{\text{th}}}$ interval
$\widehat{E}^i$	total energy in MWh requested over the $i^{\underline{\text{th}}}$ interval
$\overline{E}_j$	upper bound of energy production for unit j
$\underline{E}_j$	lower bound of energy production for unit j
f_j	$=\widetilde{f}_j - \widetilde{f}_{N_u+1}$
$\widetilde{f}_j$	generation cost (linear) in €/MWh of $j^{\underline{\text{th}}}$ unit
$\widetilde{f}_{N_u+1}$	generation cost (linear) in €/MWh of external emergency (power-unlimited) energy

$\widehat{f^i}$	$f^i - A^{i}_{=}{}'\Pi^{i}_{=} - A^{i}_{\geq}{}'\Pi^{i}_{\geq} - A^{0\,i}_{=}{}'\Pi^{0}_{=} - A^{0\,i}_{\geq}{}'\Pi^{0}_{\geq}$
i	(supraindex) indication of $i^{\underline{\text{th}}}$ interval
j	(subindex) indication of $j^{\underline{\text{th}}}$ thermal unit
L^i_0	electric load to be supplied over the $i^{\underline{\text{th}}}$ interval
$L^i_j(x)$	electric load still to be supplied over the $i^{\underline{\text{th}}}$ interval after loading (in loading order) units up to the $j^{\underline{\text{th}}}$
N	(subindex) indication of nonbasic variable
$N_=$,$N_\geq$	total numbers of equality and inequality constraints
N_i,N_u,	numbers of intervals and number of thermal units
N^i_u	number of available thermal units in $i^{\underline{\text{th}}}$ interval
N^i_V	number of vertices of the polyhedron defined by the load-matching constraints of the $i^{\underline{\text{th}}}$ interval
$\widehat{P}^i$	Peak power load of $i^{\underline{\text{th}}}$ interval (MW)
$R^0_=$	right-hand sides of equalities that refer to energies of more than one interval
$R^0_\geq$	right-hand sides of inequalities that refer to energies of more than one interval
$R^i_=$	right-hand sides of equalities that refer only to energies of $i^{\underline{\text{th}}}$ interval
$R^i_\geq$	right-hand sides of inequalities that refer only to energies of $i^{\underline{\text{th}}}$ interval
S^0	surpluses of inequalities that refer to energies of more than one interval
S^i	surpluses of inequalities that refer only to energies of $i^{\underline{\text{th}}}$ interval
T	total duration of interval (h)
x	generated power (MW)
Λ^i	convex coefficients of vertices of $i^{\underline{\text{th}}}$ interval
$\Pi^0_=$	Lagrange multipliers of non-load-matching equalities refering to several intervals
$\Pi^0_\geq$	Lagrange multipliers of non-load-matching inequalities refering to several intervals
$\Pi^i_=$	Lagrange multipliers of non-load-matching equalities refering only to $i^{\underline{\text{th}}}$ interval
$\Pi^i_\geq$	Lagrange multipliers of non-load-matching inequalities refering only to $i^{\underline{\text{th}}}$ interval
π^i_λ	Lagrange multipliers of convexity condition for $i^{\underline{\text{th}}}$ interval ($\sum_k \lambda^i_k$=1)
$\sigma^0_{S\,l}$	Lagrange multiplier of nonbasic surplus of multi-interval inequality l ($s^0_{N\,l}$=0)
$\sigma^i_{S\,l}$	Lagrange multiplier of nonbasic surplus of $i^{\underline{\text{th}}}$ interval inequality l ($s^i_{N\,l}$=0)
$\sigma^i_{V\,k}$	Lagrange multiplier of nonbasic convex coefficient of vertex k of $i^{\underline{\text{th}}}$ interval ($\lambda^i_{N\,k}$=0)

References

[1] H. Balériaux, E. Jamoulle, and F. Linard de Gertechin. 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, 1967.

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

[3] A.J. Conejo and Prieto. Mathematical programming and electricity markets. *TOP*, 9(1):1–53, 2001.

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

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

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

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

[8] N. Nabona and M.A. Díez. Long-term thermal multi-interval thermal generation scheduling using Bloom and Gallant's linear model. Technical Report DR9803, Dept. Estadística i Inv. Operativa, Univ. Politèc. de Catalunya, 08028 Barcelona, 1998.

[9] N. Nabona, C. Gil, and J. Albrecht. 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, 2001.

[10] N. Nabona and A. Pagès. 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. Politèc. de Catalunya, 08028 Barcelona, 2002.

[11] A. Pagès. 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, 2002.

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

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