

DIPLOMATURA D'ESTADÍSTICA

Títol: *Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric*
Alumne : Albert Roso Llorach
Directors: F. J. Heredia
C. Corchero
Data: Setembre 2007

UNIVERSITAT POLITÈCNICA DE CATALUNYA ^L
Biblioteca



1400697779



Facultat de Matemàtiques
i Estadística

Índex

Índex	2
1- Introducció	4
1.1 Descripció del PFC	5
1.2 Objectius del PFC	5
1.3 Planificació del PFC	6
1.4 Resultats esperats	6
1.5 Contingut de la memòria	6
2- Optimització de l'oferta del mercat elèctric	7
2.1 Mercat Ibèric de l'Electricitat	8
2.1.1 Introducció al Mercat Ibèric de l'Electricitat	8
2.1.2 Introducció al preu de l'electricitat	10
2.2 Modelització del problema d'optimització	13
2.2.1 Introducció a la Programació Estocàstica	13
2.2.2 Arbre d'escenaris	14
2.2.3 Model d'optimització	16
3- Tècniques de reducció d'escenaris	18
3.1 Descripció dels mètodes utilitzats	19
3.1.1 Nomenclatura	19
3.1.2 Fonaments teòrics	20
3.1.3 Algorismes	21
3.2 Llenguatge d'implementació	25
3.2.1 AMPL	25
3.2.1 Estructura del programa en AMPL	26
3.3 Generació de l'arbre d'escenaris	27
3.3.1 Elements de la implementació de l'arbre d'escenaris	27
3.3.2 Codi de l'arbre d'escenaris	30
3.4 Implementació dels algorismes	34
3.4.1 Elements de la implementació dels algorismes	34
3.4.2 Codi de Simultaneous Backward Reduction	35
3.4.3 Codi de Fast Forward Selection	37
3.5 Resultats	40
3.5.1 Variables de sortida	40
3.5.2 Codi de les variables de sortida	41
3.6 Execució	42
3.6.1 Exemple	43
4- Proves computacionals	45
4.1 Proves amb arbres d'escenaris	46
4.1.1 Cas 1	46
4.1.2 Cas 2	48
4.1.3 Cas 3	51
4.2 Aplicació al problema d'oferta òptima al mercat diari d'energia elèctrica	54
5- Conclusions	56
5.1 Resum	57
5.2 Consideracions finals	58
5.3 Propostes a desenvolupar	59
6- Bibliografia	60
7- Annexos	62
Annex 1	63

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

Annex 2.....	65
Annex 3.....	67
Annex 4.....	69

1- Introducció

1.1 Descripció del PFC

El Projecte Fi de Carrera presentat tracta sobre la construcció d'arbres d'escenaris i la seva aplicació en problemes de Programació Estocàstica.

Un arbre d'escenaris constitueix una representació discreta del conjunt de possibles estats futurs d'un procés estocàstic, per exemple, la càrrega elèctrica, el preu de l'electricitat, el preu del fuel, etc.

Normalment els arbres generats contenen un nombre d'escenaris massa gran, fet que comporta una costosa i poc eficient resolució dels models d'optimització on són utilitzats. Per tal d'aconseguir una eficient resolució, duem a terme una aproximació de l'arbre original amb un arbre format per un nombre més reduït d'escenaris.

Per tal de poder realitzar aquesta reducció s'han implementat dos tipus d'algorismes de reducció d'escenaris descrits en l'article de Gröwe-Kuska, Heitsch i Römisch [10]:

- Simultaneous Backward Reduction
- Fast Forward Selection

Es tracta de dos algorismes heurístics de reducció que determinen un subconjunt del conjunt d'escenaris inicials i assignen noves probabilitats als escenaris conservats.

La metodologia de *Simultaneous Backward Reduction* es basa en l'eliminació d'escenaris fins a que resten el nombre desitjat d'escenaris conservats. Mentre que en el cas de *Fast Forward Selection* es fonamenta en la selecció d'escenaris fins a obtenir el nombre desitjat d'escenaris preservats.

Aquests dos algorismes han estat implementats en el llenguatge de modelització matemàtica AMPL.

Els nous arbres reduïts, seran utilitzats en la resolució d'un problema d'optimització de l'oferta al mercat elèctric diari.

1.2 Objectius del PFC

En la realització d'aquest PFC es van proposar assolir els següents objectius:

- Estudi i comprensió de les tècniques de reducció d'escenaris proposades a [10].
- Estudi i comprensió dels aspectes bàsics del funcionament del Mercat Ibèric de l'Electricitat.
- Implementació de les tècniques de reducció d'escenaris en AMPL.
- Aplicació de les tècniques de reducció d'escenaris als preus del mercat elèctric.
- Estudi computacional del comportament dels algorismes de reducció.

- Ús dels arbres generats en un model d'optimització de l'oferta al mercat elèctric diari.

1.3 Planificació del PFC

Es va planificar el projecte en els següents terminis:

Mes	Tasca
1	Aprentatge llenguatge AMPL Assolir coneixements Programació Estocàstica i arbres d'escenaris
2	Implementació Arbre d'escenaris + millora
3	Implementació Algorisme 1 + millora
4	Implementació Algorisme 2 + millora
5	Proves Computacionals + Model Optimització.
6	Redacció memòria PFC

Taula 1.1: Planificació PFC per mesos

1.4 Resultats esperats

Els resultats que s'esperaven eren:

- Implementacions dels dos algorismes presentats anteriorment que obtinguessin uns arbres d'escenaris reduïts que representessin al màxim els arbres originals.
- Una implementació eficient dels mètodes plantejats.
- Amb aquests resultats, obtenir solucions satisfactòries per a un model d'optimització de l'oferta al mercat elèctric diari.

1.5 Contingut de la memòria

L'estructura del projecte és la següent:

- Apartat 2. Optimització de l'oferta del mercat elèctric: introducció al Mercat Ibèric de l'Electricitat. Fonaments de Programació Estocàstica i arbres d'escenaris. Model d'optimització de l'oferta al mercat elèctric diari.
- Apartat 3. Tècniques de reducció d'escenaris: descripció de les tècniques emprades en la reducció d'escenaris i implementació en el llenguatge de modelització matemàtica AMPL.
- Apartat 4. Proves computacionals: tests computacionals de la implementació dels algorismes de reducció d'escenaris i resultats del model d'optimització utilitzant aquests mètodes.
- Conclusions

2- Optimització de l'oferta del mercat elèctric

2.1 Mercat Ibèric de l'Electricitat

2.1.1 Introducció al Mercat Ibèric de l'Electricitat

A partir de l'1 de gener de 1998 [1], l'organització del sector elèctric espanyol pateix una sèrie de canvis que porten a la liberalització de la producció i comercialització d'electricitat. L'explotació elèctrica s'organitzarà a partir d'aquest moment al voltant del Mercat Elèctric Espanyol, on les companyies generadores vendran la seva energia i les companyies distribuïdores l'adquiriran.

Aquest nou model d'organització suposa un canvi al funcionament del sector, s'abandona la noció de servei públic mantenint, però, la garantia de subministrament a tots els consumidors. Es reconeix al sector de l'energia elèctrica el dret a la lliure instal·lació de generadors i s'organitza el seu funcionament sota el principi de lliure competència amb la creació d'un mercat liberalitzat d'electricitat on els preus es fixen en funció de les decisions adoptades lliurement pels agents participants.

Més tard, a partir del 20 d'abril de 2004 comença a funcionar el Mercat Ibèric de l'Electricitat (MIBEL) que és el resultat de la integració dels mercats d'electricitat espanyol i portuguès.

Per assegurar el correcte funcionament del sistema s'estableixen les figures dels Operadors de Mercat, l'Operador del Sistema i el Regulador, amb les següents funcions:

- Operadors de Mercats (Operador del Mercado Eléctrico (OMEL) [2] i Operador del Mercado Ibérico de Portugal (OMIP) [3]):
 - Gestió econòmica del Mercat diari (OMEL).
 - Gestió econòmica del Mercat de futurs (OMIP).
 - Tramitació de les ofertes de compra/venda, cassació de les mateixes i comunicació dels resultats.
 - Liquidació i comunicació de pagaments i cobraments
 - Normativa i regulació del Mercat de producció
- Operador del Sistema (Red Eléctrica Española[4]):
 - Gestió tècnica del sistema, garantint la seguretat, fiabilitat i qualitat del subministrament
 - Garantir l'equilibri entre la producció i la demanda
 - Normativa i regulació dels procediments d'operació tècnica del sistema
- Regulador (Comisión Nacional de Energía [5]):
 - Garantir el funcionament del Mercat en base a la lliure competència
 - Evitar abusos de posicions de domini
 - Realitzar els anàlisis de planificació elèctrica

- Establir les retribucions de les activitats regulades
- Fixar tarifes
- Determinar els requisits mínims de qualitat i seguretat del servei

El Mercat Elèctric es compon de diferents Mercats ordenats temporalment, el mercat diari, el mercat de secundària i els mercats intradiaris. El mercat diari fa una primera aproximació dels preus i de les ofertes acceptades, que després s'acabaran de definir amb ajustos realitzats pels altres mercats segons factors com les restriccions tècniques, els contractes bilaterals, ...

És al Mercat diari on es porten a terme les transaccions de compra – venda d'energia elèctrica per el dia següent, dia de subministrament. Les companyies, productores i distribuïdores, han d'enviar ofertes horàries simples, indicant quantitat oferta i el preu corresponent, de compra i venda respectivament. En aquest mercat han de participar com a oferents totes les unitats de producció disponibles i com a compradors totes les que vulguin tenir dret a participar als mercats intradiaris.

El següent gràfic (figura 2.1) mostra les ofertes presentades pels grups generadors, ordenades segons la quantitat d'energia oferta. A partir de les ofertes de les companyies distribuïdores es fixa la demanda del sistema a la hora considerada, i la intersecció entre aquests valors ens proporciona el preu marginal per aquest interval.

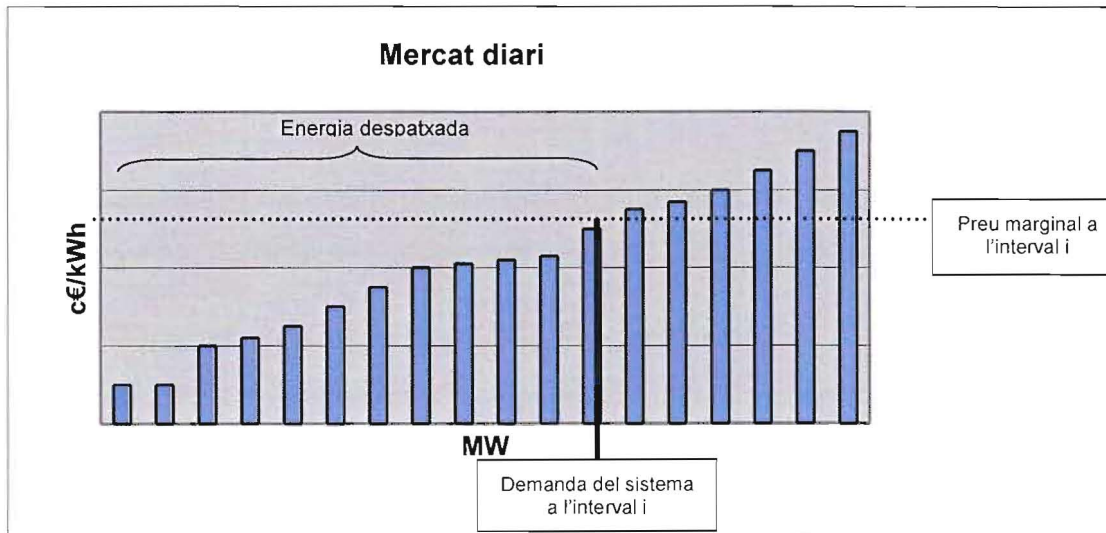


Figura 2.1: Ofertes presentades pels grups generadors

A partir d'aquestes ofertes es construeixen les corbes d'oferta (figura 2.2) i mitjançant la cassació es fixa el preu marginal, que ve donat per la darrera oferta de venda acceptada. Totes les transaccions es liquidaran al preu marginal de l'hora considerada. El procediment de cassació es duu a terme tenint en compte tota la informació de la que disposa l'Operador del Sistema: indisponibilitats, contractes bilaterals, contractes internacionals, ofertes de venda i ofertes de compra.

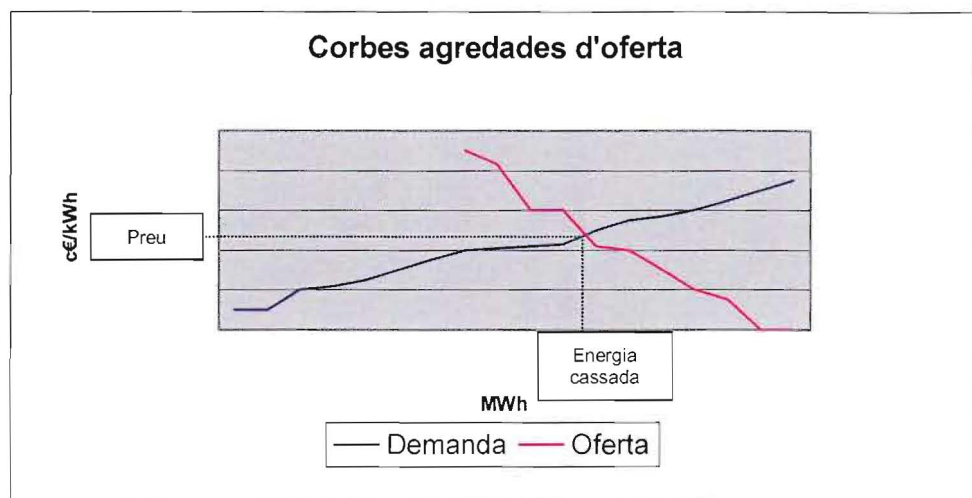


Figura 2.2: Corbes agregades d'oferta

Per tant, existeixen diversos factors que són font d'incertesa i que condicionaran els preus de mercat:

- Factors derivats del propi comportament de factors exògens:
 - Combustibles
 - Consum específic
 - Demanda
 - Hidraulicitat
 - Disponibilitat
- Factors derivats del comportament dels participants al Mercat
- Factors derivats de l'estructura de producció i generació dels diferents competidors
- Factors derivats de l'evolució de l'entorn regulatori

El preu final de l'energia incorporarà el preu obtingut de la cassació al mercat diari, el preu de les desviacions derivades de les restriccions tècniques i el preu obtingut de la cassació als mercats intradiaris.

2.1.2 Introducció al preu de l'electricitat

El comportament del preu de mercat de la producció energètica ha estat estudiat [6], bàsicament, des del punt de vista de les sèries temporals [7], intentant fer una previsió amb un marge d'error limitat del preu segons l'històric observat.

Aquest enfocament es basa, principalment, en la forta relació existent entre el preu i el consum d'energia, essent aquest últim una variable amb un comportament típic de sèrie temporal. El consum energètic depèn quasi exclusivament de l'hora del dia, el tipus de dia i el mes en els que s'observi, existint un factor additiu que fa augmentar la mitjana anualment.

Les figures 2.3 i 2.4 representen aquestes dues sèries, consum en megawatt hora (MWh) i preu en cèntims d'euro per kilowatt hora (c€/kWh), per l'any 2003, es pot observar que la sèrie del consum té una estructura regular i estacional, mentre que la corba del preu presenta més variabilitat. Aquesta situació es repeteix per tots els anys dels que es disposen dades.

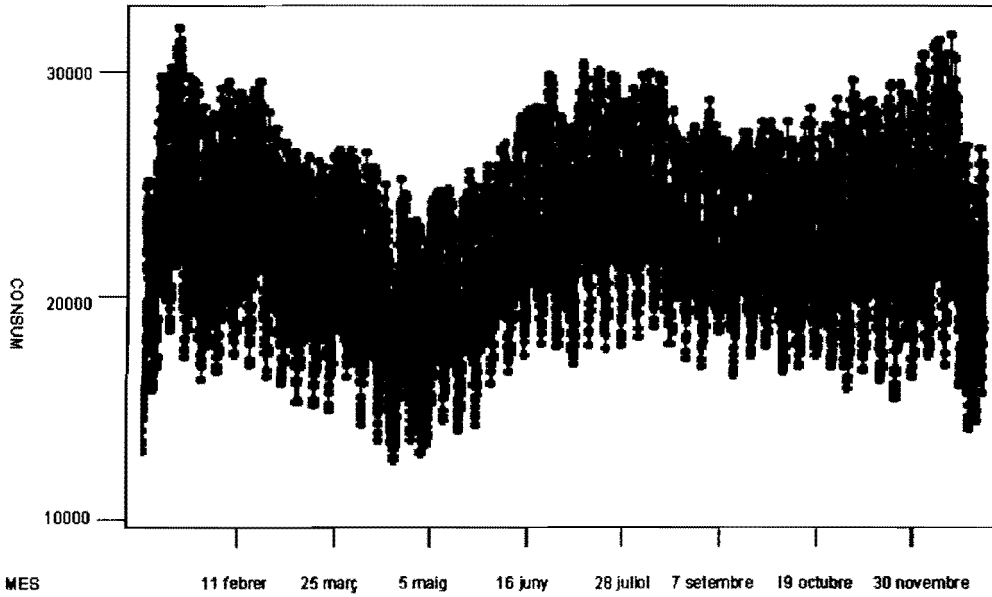


Figura 2.3: Sèrie del preu any 2003

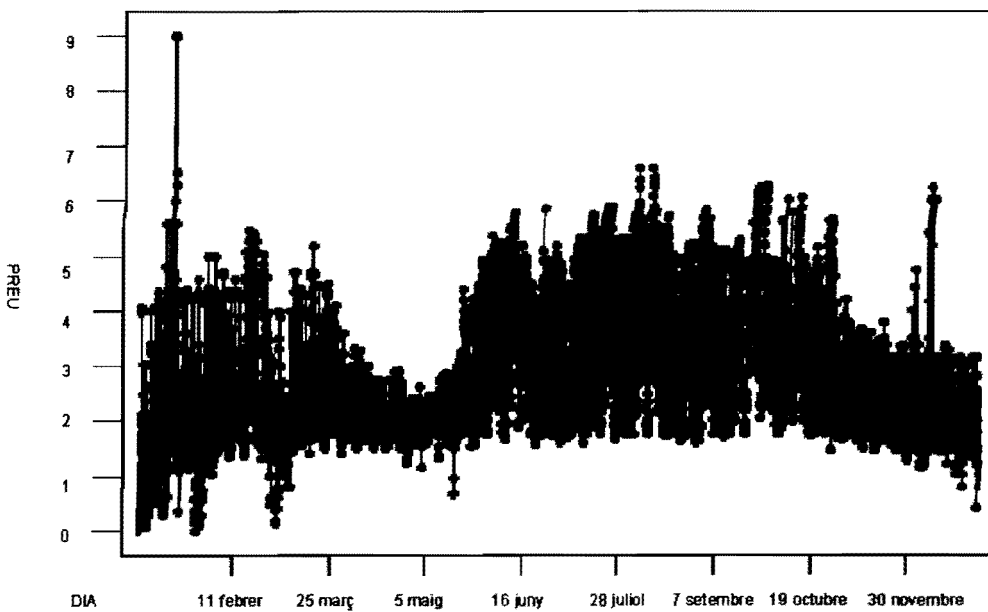


Figura 2.4: Sèrie del preu any 2003

A la figura 2.5 cada línia representa la mitjana del consum per cada hora d'un determinat mes. Es pot observar que presenta una estructura diària que es podria descriure com super vall, punta, vall, super punta, trobant-se el màxim del consum diari al voltant de les 21 hores. Aquesta estructura es repeteix, amb lleugeres diferències, tots els dies de l'any, traslladant-la al llarg de l'eix de la quantitat consumida segons el mes en estudi.

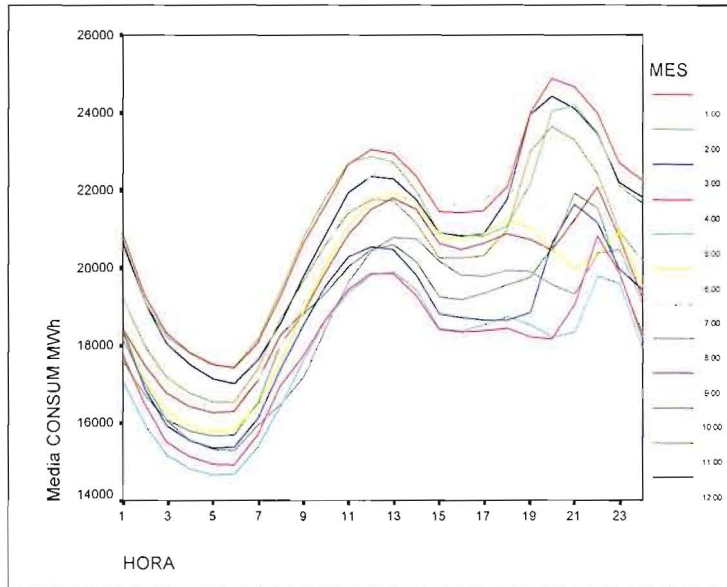


Figura 2.5: Corba consum mitjà horari pels diferents mesos

A la figura 2.6 cada línia representa la mitjana del preu de l'energia per cada hora per un determinat mes. Es pot observar que presenta una estructura molt similar a la del consum però amb molta més variabilitat.

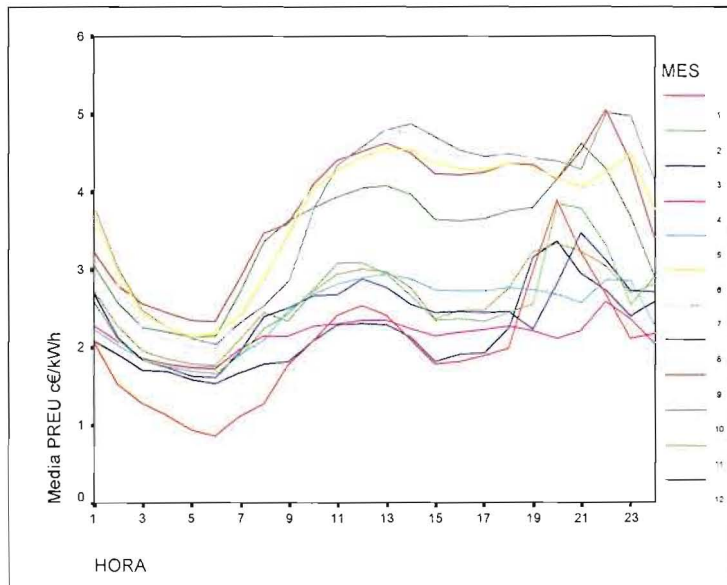


Figura 2.6: Corba mitjana preu horari pels diferents mesos

2.2 Modelització del problema d'optimització

2.2.1 Introducció a la Programació Estocàstica

La tècnica coneguda com *programació estocàstica (stochastic programming)* o *optimització amb incertesa (optimization under uncertainty)* pretén trobar la decisió òptima a problemes que inclouen incertesa en les seves dades [8]. Habitualment aquesta incertesa es tracta transformant les dades fins obtenir un problema determinista. En el cas de la programació estocàstica s'aprofita aquesta incertesa dins el model, fent-ho, en principi, més proper al comportament real del sistema.

Quan es diu que existeix incertesa a les dades es suposa que aquestes poden ser representades mitjançant variables aleatòries amb distribució coneguda. Les dades reals només es coneixeran un cop ha passat la realització de la variable aleatòria.

Les variables de decisió d'un model de programació estocàstica es divideixen en dos grups:

- variables de primera etapa: són aquelles que corresponen a decisions que han de ser preses abans de que es conegui la realització de la variable aleatòria. El període en el que es prenen es coneix com primera etapa.
- variables de segona etapa: són aquelles que depenen de la realització de la variable aleatòria i per tant només poden ser preses després de la realització d'aquesta. El període corresponent s'anomena segona etapa.

La formulació d'un problema estocàstic lineal de dos etapes amb recurs fix és:

$$\begin{aligned} \min z &= c'x + E_{\xi}[\min q(\omega)'y(\omega)] \\ \text{s.t. } Ax &= b \\ T(\omega)x + Wy(\omega) &= h(\omega) \\ x \geq 0; y &\geq 0 \end{aligned} \tag{2.1}$$

on les decisions de primera etapa venen representades per les variables x i els vectors coneguts c a \mathbb{R}^n , b a \mathbb{R}^m i A a $\mathbb{R}^{m \times n}$. La segona etapa ve representada per les variables $y(\omega)$, la matriu W a $\mathbb{R}^{m_2 \times n_2}$, coneguda com la *matriu de recurs* i que es considera fixa, i els vectors $q(\omega)$ a \mathbb{R}^{n_2} , $h(\omega)$ a \mathbb{R}^{m_2} i $T(\omega)$ a $\mathbb{R}^{m_2 \times n_2}$ dependents de la variable aleatòria ξ . E_{ξ} representa l'esperança matemàtica respecte la variable aleatòria ξ .

La funció objectiu (2.1) conté un terme determinista, $c'x$, i l'esperança de la funció objectiu de la segona etapa, $q(\omega)'y(\omega)$, tenint en compte totes les possibles realitzacions de la variable aleatòria. Aquesta formulació fa que les decisions de primera etapa es prenguin tenint en compte la incertesa de la futura realització de ξ .

A la segona etapa, un cop observada la realització ω de la variable aleatòria ξ , es coneixen $q(\omega)$, $h(\omega)$ i $T(\omega)$ i és llavors quan s'ha de prendre la decisió corresponent a

la variable $y(\omega)$. Per tant, les decisions de primera etapa es prenen tenint en compte el seu futur efecte.

Per aquesta formulació es considera que la variable aleatòria ξ és discretitzada, poden prendre un número finit de escenaris s , representacions de la possible evolució d'un sistema cap a un estat futur, amb una certa probabilitat P_s . A partir d'aquesta premissa es desdoblen les variables de segona etapa y , passant a tenir una per cada escenari, y^s , i es pot definir el problema de la forma anomenada extensa :

$$\begin{aligned} \min z &= c'x + \sum_{s=1}^S P_s (q(\omega)_s' y(\omega)_s) \\ \text{s.t. } Ax &= b \\ \left. \begin{aligned} T(\omega)_s x + W y(\omega)_s &= h(\omega)_s \\ y_s &\geq 0 \\ x &\geq 0 \end{aligned} \right\} \forall s \in S \end{aligned} \quad (2.2)$$

2.2.2 Arbre d'escenaris

La incertesa d'un problema estocàstic es pot representar mitjançant un arbre amb un nombre finit d'escenaris [9], de forma que el problema estocàstic es converteix en un determinista.

Un arbre de probabilitats o escenaris constitueix una representació discreta dels possibles estats o nodes $\omega \in \Omega$ de l'arbre que poden prendre les variables aleatòries del model en cada període $t \in T$ de l'abast del problema.

Qualsevol camí o trajectòria que va des de l'arrel de l'arbre fins a les fulles és el que es defineix com a escenari. Aquells que comparteixen una mateixa informació fins a una certa etapa també comparteixen aquestes branques de l'arbre.

La probabilitat de transició d'un node a un altre es representa per p^ω , mentre que p^s és la probabilitat de que succeeixi l'escenari s . Existeix una correspondència unívoca entre cada un dels nodes de l'arbre d'escenaris i el seu predecessor, de forma que si E_s és el conjunt de nodes ω que pertanyen a l'escenari s , ambdues probabilitats estan lligades per l'expressió:

$$p^s = \prod_{\omega \in E_s} p^\omega \quad (2.3)$$

Cada node pot ramificar-se un nombre indefinit de vegades, encara que normalment aquestes ramificacions són reduïdes ja que eleven considerablement la mida dels problemes.

Una de les estructures més habituals d'arbre és aquella amb bifurcacions en cada node. En la figura 2.7 es representa un arbre d'aquest tipus format per vuit escenaris $E = \{E_{s_1}, \dots, E_{s_8}\}$, tres períodes $T = \{t_1, t_2, t_3\}$ i catorze nodes $\Omega = \{\omega_1, \dots, \omega_{14}\}$. El primer escenari, per exemple, està compost pels nodes $E_{s_1} = \{\omega_1, \omega_3, \omega_7\}$ i té una probabilitat de succeir de $p^{s_1} = p^{\omega_1} \times p^{\omega_3} \times p^{\omega_7}$.

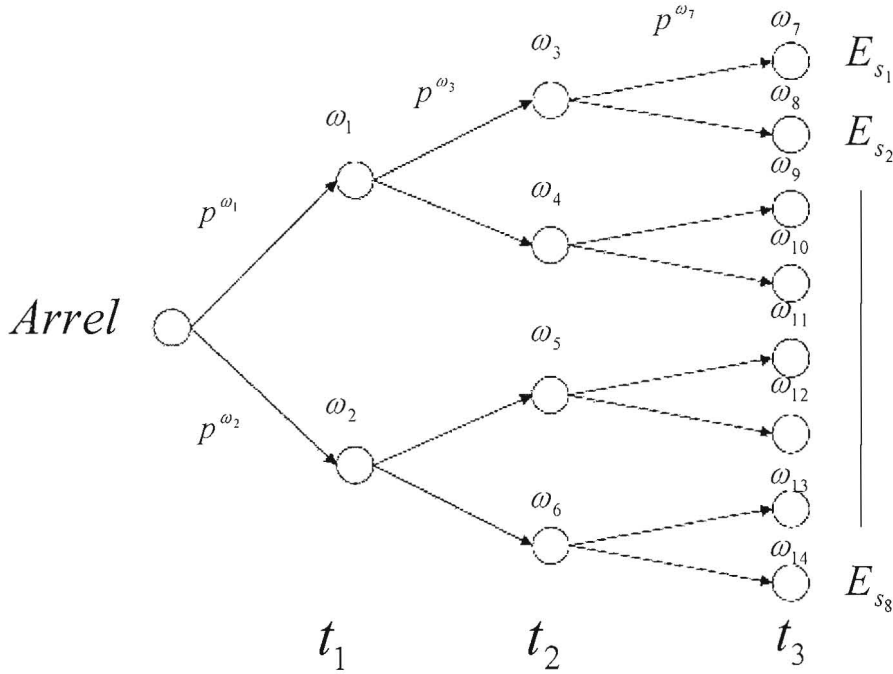


Figura 2.7: Arbre d'escenaris amb bifurcacions a cada node

Adicionalment, per a recórrer l'arbre es defineix el conjunt Ω_t de nodes ω que pertanyen al període t . Així, el conjunt de nodes del primer període es $\Omega_1 = \{\omega_1, \omega_2\}$.

En la Figura 2.8 s'observa el cas particular de representació d'un arbre d'escenaris amb una branca per a cada node i amb probabilitats p^s equiprobables, anomenat arbre fan.

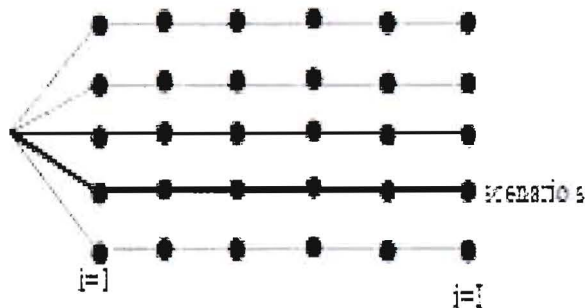


Figura 2.8: Arbre d'escenaris tipus fan

2.2.3 Model d'optimització

Com s'ha comentat en apartats anteriors, el model que es presenta a continuació respon a les noves necessitats d'una companyia generadora elèctrica a l'hora de planificar la seva producció dins dels mercats energètics. Es presenta un model simplificat creat amb l'objectiu d'il·lustrar la necessitat de la construcció dels arbres d'escenaris.

Suposem que cada companyia disposa d'un conjunt de contractes de futurs, F , signats amb antelació a la cassació del Mercat Diari. Això implica que la companyia coneix la quantitat mínima de producció que ha de cassar al mercat, que serà la suma de les quantitats de cada contracte de futur, L_f . Per tant, la companyia ha de decidir com produir de forma òptima a través del parc d'unitats tèrmiques aquesta quantitat. Per garantir que aquesta quantitat entra en la cassació del mercat s'ofereix a preu instrumental de venda (0€/MW) dins de l'oferta de cada unitat.

Es formula de forma extensa un problema estocàstic amb recurs fix, funció objectiu quadràtica i constriccions lineals. La variable aleatòria representa el preu del mercat diari, λ , i s'ha d'introduir al model mitjançant un arbre d'escenaris. Les variables de decisió de primera etapa són l'oferta a preu instrumental de venda q_t^i per cada unitat t i interval i , i $q_{if}^{F,t}$ la part d'aquesta oferta assignada a cada contracte de futur f . Les variables de segona etapa, que depenen de l'escenari considerat, són la potència total cassada, p_t^s per cada unitat t , interval i i escenari s , és a dir, el la quantitat total d'energia que hauria de produir la unitat t si el preu de mercat a l'interval i fos exactament el predit per l'escenari s .

Notació:

- Conjunts:

- I : nombre total d'interval·ls del dos períodes (48 en el nostre cas).
- I^1 : nombre total d'interval·ls del primer període (24 en el nostre cas).
- F : nombre de contractes de futurs.
- S : nombre d'escenaris
- T_{on} : conjunt de parell ordenats (t,i) corresponents a tèrmiques t obertes durant el període i .
- R : nombre de conjunts d'escenaris
- $B(r)$: conjunt ordenat d'escenaris que pertanyen al grup r , $r = 1, \dots, R$.

- Paràmetres:

- L_f : quantitat d'energia a produir pel contracte f , $f = 1, \dots, F$.
- P_s : probabilitat de l'escenari s , $s = 1, \dots, S$.
- λ_i^s : preu de mercat per l'escenari s , interval i , $\forall s = 1, \dots, S, \forall i = 1, \dots, I$.
- \bar{P}^t : potència màxima unitat t , $\forall t \in T_{on}$.
- \underline{P}^t : potència mínima unitat t , $\forall t \in T_{on}$.

c_i^t : terme lineal de la funció de costos de producció unitat t , $\forall t \in T_{on}$.

c_q^t : terme quadràtic de la funció de costos de producció unitat t , $\forall t \in T_{on}$.

Model:

$$\min \sum_{(t,i) \in T_{on}} \sum_{s=1}^S P_s ((c_i^t - \lambda_i^s) p_i^{ts} + c_q^t (p_i^{ts})^2)$$

$$s.a. \quad p_i^{ts} \geq q_i^t \quad \forall (t,i) \in T_{on}, s = 1, \dots, S \quad (a)$$

$$q_i^t \geq \sum_{f \in F} q_{i,f}^{Ft} \quad \forall (t,i) \in T_{on} \quad (b)$$

$$\sum_{(t,i) \in T_{on}} q_{i,f}^{Ft} = L_j \quad i = 1, \dots, I, f = 1, \dots, F \quad (c) \quad (2.4)$$

$$p_i^j = p_i^k \quad \begin{cases} \forall j, k \in B(r), j \neq k, r = 1, \dots, R \\ \forall (t,i) \in T_{on}, i = 1, \dots, I \end{cases} \quad (d)$$

$$p_i^{ts} \leq \bar{P}^t \quad \forall (t,i) \in T_{on}, s = 1, \dots, S \quad (e)$$

$$q_i^t \geq \underline{P}^t \quad \forall (t,i) \in T_{on} \quad (f)$$

La funció objectiu minimitza la esperança de la diferència entre els costos de producció i els beneficis obtinguts de la venda de l'energia al mercat.

Les constriccions defineixen la relació entre l'energia total cassada i l'oferta a preu instrumental (a), relacionen l'oferta a preu instrumental total i per cada unitat (b) i garanteixen la producció del total de l'energia dels contractes de futurs (c). El conjunt (d) són les constriccions de no-anticipativitat, i cal imposar-les per tal que les variables de decisió de segona etapa associades a la mateixa branca del primer període tenen el mateix valor, i modelitzen el fet que les decisions preses durant el primer període només poden tenir en compte la informació disponible en aquell moment [8]. Per últim s'afiten les variables de producció, (e) i (f).

3- Tècniques de reducció d'escenaris

3.1 Descripció dels mètodes utilitzats

Habitualment, les tècniques utilitzades per a la construcció d'arbres obtenen uns arbres d'escenaris massa grans, fet que comporta una costosa i poc eficient resolució del problema d'optimització. Per aquest motiu, es porta a terme una aproximació de l'arbre original amb un arbre amb un nombre més reduït d'escenaris.

Per a poder realitzar la reducció de l'arbre d'escenaris, s'han utilitzat els mètodes plantejats en l'article de Gröwe-Kuska, Heitsch i Römisch [10] per a reducció d'escenaris.

3.1.1 Nomenclatura

S'ha de considerar que els mètodes presentats no requereixen cap tipus d'imposició sobre les dades del procés observat (la dimensió del procés o una estructura de dependència) o sobre l'estructura dels escenaris (estructurats en arbre o no).

Definim:

- $\xi, \{\xi_t\}_{t=1}^T$ Processos estocàstics de dimensió n amb $t \in \{1, \dots, T\}$, on T és el nombre total d'etapes.
- $\tilde{\xi}, \{\tilde{\xi}_t\}_{t=1}^T$ Escenaris que pertanyen a $\xi, \tilde{\xi}$.
- p_i, q_j Probabilitats dels escenaris, amb $p_i \geq 0, q_j \geq 0, \sum_i p_i = \sum_j q_j = 1$.
- P, Q Distribució probabilística dels processos $\xi, \tilde{\xi}$, respectivament.

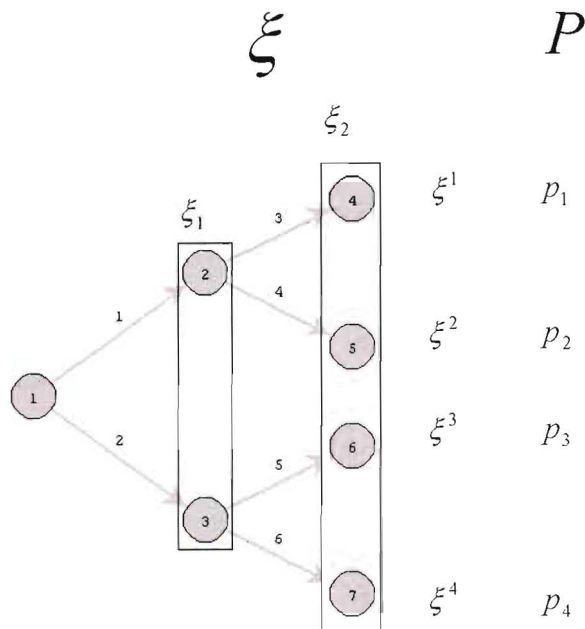


Figura 3.1: Representació d'un arbre d'escenaris de dues etapes

S	Nombre d'escenaris que existeixen en el conjunt inicial d'escenaris.
J	Conjunt dels índexs dels escenaris eliminats.
$\#J$	Cardinalitat del conjunt J , és a dir el nombre d'escenaris eliminats.
$s = S - \#J$	Nombre d'escenaris preservats.
ε	Tolerància de la reducció.
$c_t(\xi^i, \xi^j)$	Distància entre els escenaris $\{\xi^i\}_{t=1}^t, \{\xi^j\}_{t=1}^t$.

3.1.2 Fonaments teòrics

Assumim que la distribució de probabilitat P del procés estocàstic de dimensió n

$\xi = \{\xi_t\}_{t=1}^T$ (en el nostre cas preu de l'electricitat) està aproximada per un nombre finits d'escenaris $\xi^i = \{\xi_t^i\}_{t=1}^T, i = 1, \dots, S$ i les seves probabilitats $p_i, \sum_{i=1}^S p_i = 1$.

Els algorismes de reducció d'escenaris determinen un subconjunt d'escenaris, la cardinalitat del qual ha estat fixada anteriorment, i assignen noves probabilitats als escenaris que han estat preservats de forma que la distribució de probabilitat reduïda Q sigui més propera a la distribució original P en termes d'una certa distància de probabilitat entre P i Q . Aquesta distància de probabilitats té en compte les probabilitats dels escenaris i la distància dels valors dels escenaris. En el context de programació estocàstica per a problemes de gestió de l'energia s'utilitza la distància de Kantorovich D_k de distribucions multivariants de probabilitat.

Sigui Q la distribució d'un altre procés estocàstic de dimensió n $\tilde{\xi}$ amb escenaris $\tilde{\xi}^j \in R^{nT}$ i probabilitats $q_j, j = 1, \dots, \tilde{S}$. Llavors:

$$D_k(P, Q) = \inf \left\{ \sum_{i=1}^S \sum_{j=1}^{\tilde{S}} \eta_{ij} c_T(\xi^i, \tilde{\xi}^j) : \eta_{ij} \geq 0, \sum_{i=1}^S \eta_{ij} = q_j, \sum_{j=1}^{\tilde{S}} \eta_{ij} = p_i, \forall i, \forall j \right\}$$

on $c_t(\xi^i, \tilde{\xi}^j) := \sum_{\tau=1}^t |\xi_\tau^i - \tilde{\xi}_\tau^j|, t = 1, \dots, T$ i $\|\cdot\|$ denota algun tipus de norma a R^{nT} , en aquest cas c_T mesura la distància entre escenaris per a totes les etapes T .

Ara, sigui Q la distribució de probabilitat reduïda de ξ . Tenim que el suport de Q consisteix en els escenaris ξ^j amb $j \in \{1, \dots, S\} \setminus J$ on J denota el conjunt d'índexs dels escenaris eliminats. Per a un conjunt $J \subset \{1, \dots, S\}$ fixat, el conjunt d'escenaris de Q basat en els escenaris $\{\xi^j\}_{j \notin J}$ que tenen distància D_k mínima a P ha de ser calculat explícitament. La *distància mínima* és

$$D_k(P, Q) = \sum_{i \in J} p_i \min_{j \notin J} c_T(\xi^i, \xi^j) \quad (3.1)$$

i la probabilitat q_j dels escenaris preservats $\xi^j, j \notin J$ de Q ve donada per la regla

$$q_j := p_j + \sum_{i \in J(j)} p_i \quad (3.2)$$

on $J(j) := \{i \in J : j = j(i)\}$, $j(i) \in \arg \min_{j \in J} c_T(\xi^i, \xi^j)$, $\forall i \in J$.

La interpretació de la *regla de redistribució òptima* és que la nova probabilitat d'un escenari preservat és igual a la suma de la seva antiga probabilitat i totes les probabilitats dels escenaris eliminats que el tenen com a escenari més proper respecte c_T . Tots els escenaris eliminats tenen probabilitat zero.

Per a realitzar una selecció òptima del conjunt d'índexs J per a reducció d'escenaris amb cardinalitat fixada $\#J$ ve donada per la solució al *problema òptim de reducció*

$$\min \left\{ \sum_{i \in J} p_i \min_{j \in J} c_T(\xi^i, \xi^j) : J \subset \{1, \dots, S\}, \#J = S - s \right\} \quad (3.3)$$

on $s = S - \#J > 0$ és el nombre d'escenaris preservats.

De (3.1) i (3.3) podem deduir l'*estratègia de reducció màxima* per a determinar una distribució de probabilitat reduïda Q de ξ tal que el conjunt d'escenaris eliminats té cardinalitat màxima i la $D_k(P, Q) < \varepsilon$ és manté. En aquest cas Q és pròxima a la distribució original P amb una precisió $\varepsilon > 0$:

Estratègia de reducció màxima

Determina un conjunt d'índexs J amb cardinalitat $\#J$ com aquest:

$$\sum_{i \in J} p_i \min_{j \in J} c_T(\xi^i, \xi^j) \leq \varepsilon$$

La *regla de redistribució òptima* dona les probabilitats $q_j, j \in J$, dels escenaris preservats.

3.1.3 Algorismes

Els dos algorismes heurístics de reducció plantejats determinen un subconjunt del conjunt d'escenaris inicials i assignen noves probabilitats als escenaris conservats.

Algorisme 1 - Simultaneous Backward Reduction

- *Eliminació d'un escenari*

Si $\#J = 1$, és a dir, eliminar un sol escenari, llavors el *problema òptim de reducció* (3.3) pren la forma:

$$\min_{l \in \{1, \dots, S\}} p_l \min_{j \neq l} c_T(\xi^l, \xi^j)$$

Si el mínim s'obté a $l_* \in \{1, \dots, S\}$, l'escenari ξ^{l_*} és eliminat. La *regla de redistribució òptima* (3.2) dóna la distribució de probabilitat reduïda Q .

Si $j_* \in \arg \min_{j \neq l_*} c_T(\xi^{l_*}, \xi^j)$, llavors $q_{j_*} = p_{j_*} + p_{l_*}$ i $q_l = p_l$ per a tot $l \notin \{l_*, j_*\}$.

- Cas general

La idea fonamental d'aquest algorisme es basa en el mètode d' *Eliminació d'un escenari*, el qual ha de ser repetit fins a obtenir un nombre fixat $S - s$ d'escenaris eliminats.

0) Calcular les distàncies entre parells d'escenaris

$$c_{kj} := c_T(\xi^k, \xi^j), k, j = 1, \dots, S.$$

Ordenar resultats $\{c_{kj} : j = 1, \dots, S\}, k = 1, \dots, S$

1) Calcular el valor de la distància mínima de cada escenari l .

$$c_{ll}^{[1]} := \min_{j \neq l} c_{lj}, l = 1, \dots, S$$

Calcular el valor d'aquesta distància mínima per la probabilitat de l'escenari l .

$$z_l^{[1]} := p_l c_{ll}^{[1]}, l = 1, \dots, S.$$

Triar l'índex del valor mínim de z

$$l_1 \in \arg \min_{l \in \{1, \dots, S\}} z_l^{[1]}.$$

Actualitzar el conjunt d'escenaris eliminats J

$$\text{Conjunt } J^{[1]} := \{l_1\}.$$

i) Calcular el valor de la distància mínima de cada escenari l que no estigui eliminat.

$$c_{kl}^{[i]} := \min_{j \in J^{[i-1]} \cup \{l\}} c_{kj} \text{ per a } l \notin J^{[i-1]}, k \in J^{[i-1]} \cup \{l\}$$

Calcular el valor d'aquesta distància mínima per la probabilitat de l'escenari l que no estigui eliminat.

$$z_l^{[i]} := \sum_{k \in J^{[i-1]} \cup \{l\}} p_k c_{kl}^{[i]}, l \notin J^{[i-1]}.$$

Triar l'índex del valor mínim de z , no tenint en compte els índexs eliminats.

$$l_i \in \arg \min_{l \in J^{[i-1]}} z_l^{[i]}.$$

Actualitzar el conjunt d'escenaris eliminats J

$$\text{Conjunt } J^{[i]} := J^{[i-1]} \cup \{l_i\}.$$

S-s+1) $J := J^{[S-s]}$ és el conjunt d'índexs d'escenaris eliminats. Calcular les probabilitats òptimes per als escenaris preservats usant (3.2).

Algorisme 2 - Fast Forward Selection

- Selecció òptima d'un escenari

Si $\#J = S - 1$, és a dir, preservar un sol escenari, el *problema òptim de reducció* (3.3) pren la forma:

$$\min_{u \in \{1, \dots, S\}} \sum_{i=1}^S p_i c_T(\xi^i, \xi^u)$$

Si el mínim s'obté a $u_* \in \{1, \dots, S\}$, només l'escenari ξ^{u_*} és preservat, i la *regla de redistribució òptima* (3.2) proporciona que $q_{u_*} = p_{u_*} + \sum_{i \neq u_*} p_i = 1$.

- Cas general

La idea principal d'aquest algorisme es basa en el mètode de *Selecció òptima d'un escenari* que ha de ser repetit fins a obtenir un nombre fixat s d'escenaris preservats.

0) Calcular les distàncies entre parells d'escenaris

$$c_{ku}^{[1]} := c_T(\xi^k, \xi^u), k, u = 1, \dots, S.$$

1) Calcular el sumatori del producte de les distàncies a un escenari u per la probabilitat de l'escenari u .

$$z_u^{[1]} := \sum_{\substack{k=1 \\ k \neq u}}^S p_k c_{ku}^{[1]}, u = 1, \dots, S.$$

Triar l'índex del valor mínim de z

$$u_1 \in \arg \min_{u \in \{1, \dots, S\}} z_u^{[1]}.$$

Actualitzar el conjunt d'escenaris eliminats J

$$\text{Conjunt } J^{[1]} := \{1, \dots, S\} \setminus \{u_1\}.$$

- i) Obtenir les distàncies dels escenaris a partir de comparar les distàncies actuals amb la distància de l'escenari tractat a l'escenari preservat.

$$c_{ku}^{[i]} := \min \{c_{ku}^{[i-1]}, c_{ku_{i-2}}^{[i-1]}\}, \quad k, u \in J^{[i-1]}$$

Calcular el sumatori del producte de les distàncies actualitzades d'un escenari u per la probabilitat de l'escenari u , tenint en compte els escenaris preservats.

$$z_u^{[i]} := \sum_{k \in J^{[i-1]} \setminus \{u\}} p_k c_{ku}^{[i]}, \quad u \in J^{[i-1]}.$$

Triar l'índex del valor mínim de z

$$u_i \in \arg \min_{u \in J^{[i-1]}} z_u^{[i]}.$$

Actualitzar el conjunt d'escenaris eliminats J

$$\text{Conjunt } J^{[i]} := J^{[i-1]} \setminus \{u_i\}.$$

- s+1) $J := J^{[s-s]}$ és el conjunt d'índexs d'escenaris eliminats. Calcular les probabilitats òptimes per als escenaris preservats usant (3.2).

3.2 Llenguatge d'implementació

La implementació dels dos algorismes presentats en l'apartat 3.1 s'ha dut a terme amb AMPL [11], un llenguatge de modelització algebraic per a programació matemàtica capaç d'expressar en notació algebraica programes d'optimització.

3.2.1 AMPL

AMPL, les inicials del qual corresponen a "A Mathematical Programming Language" (Llenguatge de Programació Matemàtica en català), és un extens i potent llenguatge de modelització per a Programació Lineal o No Lineal, tant amb variables discretes com contínues.

AMPL permet una notació comuna i senzilla per a formular models d'optimització. Per tal de resoldre aquests models AMPL crida una sèrie d'optimitzadors, el més apropiat per a cada tipus de problema, per a obtenir una solució.

La gran potència del llenguatge AMPL està en separar el model en si, per una banda, i, per l'altra, les dades particulars del problema [12].

Per exemple, el típic problema amb constriccions de demanda :

$$\begin{aligned} \max z &= \sum_{j=1}^n c_j x_j \\ \text{s.a.} \quad \sum_j a_{ij} x_j &\leq b_i, \forall i = 1, \dots, m \\ x_j &\geq 0, j = 1, \dots, n \end{aligned}$$

En aquest cas tenim com a dades c_j, a_{ij} i b_i , i com a variables x_j . Això en AMPL s'escriuria de la forma següent en dos fitxers, un per al model i un altre per a les dades.

```
# MODEL: EXEMPLE.MOD
# FABRICACIÓ DE n PRODUCTES AMB m MATÈRIES PRIMERES
# PARAMÈTRES DEL MODEL
param n >=0, integer ;
param m >=0, integer ;
# CONJUNTS D' INDEXS
set PRODUCTES := 1 ..n ;
set MPRIMERES := 1 ..m ;
# VARIABLES DE DECISIÓ I RESTRICCIONS NO NEGATIVITAT
var x {j in PRODUCTES } >= 0;
# MÉS PARÀMETRES DEL MODEL
param c {i in PRODUCTES };
param b {j in MPRIMERES };
param a {( i,j) in { MPRIMERES , PRODUCTES } };
# FUNCIÓ OBJECTIU DEL MODEL
maximize z : sum {j in PRODUCTES } c[j]*x[j];
# RESTRICCIONS DEL MODEL
subject to restriccio {i in MPRIMERES } :
sum {j in PRODUCTES } a[i,j]*x[j ] <= b[i];
```

I per al fitxer de dades tindriem:

```
# DADES: EXEMPLE.DAT
param n := 3;
param m := 2;
param c:=
1 4
2 6
3 7;
param a : 1 2 3:=
1 1 2 2
2 2 1 3;
param b:=
1 25
2 30;
```

Ara que tenim plantejat el model i les seves dades, per tal de resoldre'l podem crear un fitxer amb les següents instruccions, per tal de que AMPL pugui cridar a l'optimitzador adequat i obtenir la solució del problema de Programació Lineal:

```
reset;
model exemple.mod;
data exemple.dat;
option solver cplex;
solve;
display x > exemple.res;
display z > exemple.res;
close exemple.res;
quit;
```

És possible guardar les dades de la solució del problema en un fitxer per a un millor estudi.

3.2.1 Estructura del programa en AMPL

Tot i que les tècniques que implementarem no requereixen de la resolució de cap problema d'optimització, s'ha usat el llenguatge AMPL perquè les variables de sortida que obtenim en l'execució dels algorismes seran utilitzades posteriorment en la resolució de problemes d'optimització de l'oferta al mercat elèctric diari, que han estat modelitzats en AMPL.

En aquest cas, el programa constarà d'un fitxer de comandes i un altre fitxer de dades.

En el fitxer de comandes estarà programada la generació d'un arbre a partir de dades reals o simulades i dels algorismes de reducció corresponents.

Pel que fa al fitxer de dades contindrà la informació sobre l'arbre (el nombre d'etapes, nombre de branques a cada etapa, valors escenaris, probabilitats) i sobre l'algorisme (nombre escenaris eliminats).

3.3 Generació de l'arbre d'escenaris

Una part fonamental del programa és la creació d'un arbre d'escenaris, al qual s'aplicaran els mètodes de reducció descrits.

L'objectiu serà utilitzar una estructura de dades que permeti representar d'una forma senzilla i eficient els arcs i nodes que conformen un arbre.

3.3.1 Elements de la implementació de l'arbre d'escenaris

En aquest cas s'ha considerat construir un arbre simètric, és a dir, tots els nodes pare d'un mateix nivell o etapa tindran igual nombre de nodes fills o descendents. Per a descriure els escenaris s'utilitza el concepte d'arbre amb nodes predecessors.

Per a poder crear l'estructura de l'arbre s'han tingut en compte els següents paràmetres d'entrada:

```
param T integer;  
param N{1..T} integer;
```

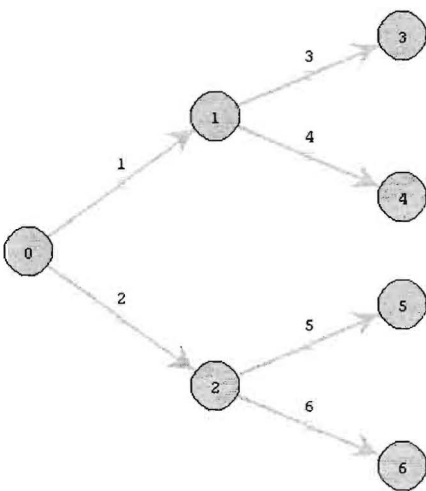
El paràmetre T correspon al nombre d'etapes que conformen l'arbre d'escenaris.
El vector N correspon al nombre de nodes fills que hi ha en cada etapa $t = 1, \dots, T$.

Els valors d'aquests paràmetres s'assignen directament del fitxer de dades.

Amb aquest valors calculem:

```
param dimp integer;  
param nod {1..T} integer;  
param S integer;
```

El paràmetre $dimp$ indica el nombre de nodes total de l'arbre.
El vector nod marca el nombre total de nodes que conté cada etapa $t = 1, \dots, T$.
El paràmetre S indica el nombre d'escenaris que estan representats en l'arbre.



En la figura 3.2 podem veure la representació d'un arbre de $T = 2$ i $N = (2,2)$. Es pot observar que amb aquest esquema el nombre de nodes serà de dimensió 6 (no tenim en compte el node arrel), el vector $nod = (2,4)$ i el nombre d'escenaris de l'arbre és $S = 4$.

Figura 3.2: Exemple d'arbre d'escenaris

Per a poder obtenir més informació sobre la representació de l'arbre s'ha creat:

```
set CN {1..T} within {1..dimp} ordered;
param cjt {1..dimp} integer;
```

El conjunt CN representa el conjunt ordenat de nodes $i = 1, \dots, dimp$ que pertanyen a l'etapa $t = 1, \dots, T$.

El vector cjt marca a quina etapa pertany el node $i = 1, \dots, dimp$.

Amb l'exemple presentat a la figura 3.2 obtindriem

```
set CN[1] := 1 2;
set CN[2] := 3 4 5 6;
```

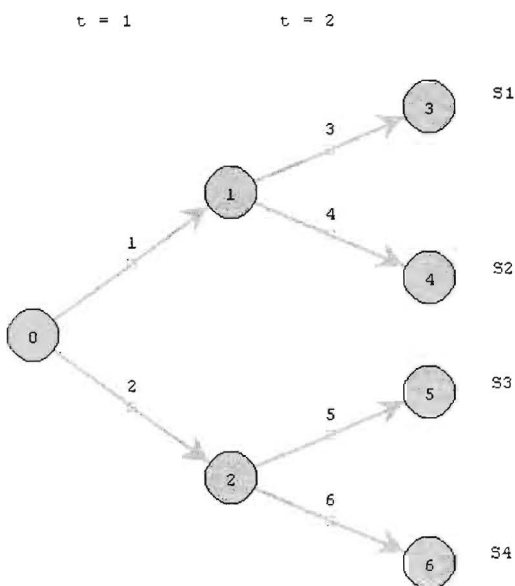
```
cjt [*] :=
1 1
2 1
3 2
4 2
5 2
6 2
;
```

A partir d'aquest coneixement s'han creat elements que representen els escenaris de l'arbre, a partir dels nodes que formen l'escenari l'escenari $i = 1, \dots, S$:

```
param A {1..T, 1..S};
set ant {1..dimp} within {1..S} ordered;
```

La matriu A indica per a cada etapa $t = 1, \dots, T$ quin és el node que pertany a l'escenari $i = 1, \dots, S$.

El conjunt ant correspon al conjunt ordenat d'escenaris $i = 1, \dots, S$ al qual el node $j = 1, \dots, dimp$ pertany.



A partir de la representació de la figura 3.3, el nostre codi obté el següent resultat per a A i ant

```
A [*,*]
: 1 2 3 4 :=
1 1 1 2 2
2 3 4 5 6
;
```

```
set ant[1] := 1 2;
set ant[2] := 3 4;
set ant[3] := 1;
set ant[4] := 2;
set ant[5] := 3;
set ant[6] := 4;
```

Figura 3.3: Exemple d'arbre d'escenaris amb etapes i escenaris

Per a poder formular les constriccions de no-anticipativitat en models d'arbres d'escenaris de dues etapes tenim els següents elements:

```
param ngrup integer;
set grup{1..ngrup}within {1..S} circular;
```

El paràmetre *ngrup* indica en quants grups s'agrupen els escenaris.

El conjunt *grup* correspon al conjunt d'escenaris $i = 1, \dots, S$ que pertanyen al grup d'escenaris $j = 1, \dots, ngrup$.

Amb les dades de la figura 3.3 tenim:

```
ngrup = 2;
set grup[1] := 1 2;
set grup[2] := 3 4;
```

Ara que ja tenim definida l'estructura de l'arbre, assignarem les probabilitat de transició d'un node a un altre, donat el fet que existeix una correspondència unívoca entre cada un dels nodes de l'arbre d'escenaris i el seu predecessor.

També per a cada node que pertany a l'arbre assignarem els seus preus associats, fet que ens permetrà construir el conjunt de preus dels escenaris:

```
param pi {1..dimp};
param prob {1..S};
param preu {1..dimp,1..24};
param E {1..S,1..24*T};
```

El vector *pi* correspon a la probabilitat de transició de cada node $i = 1, \dots, dimp$.

El vector *prob* indica la probabilitat de cada escenari $i = 1, \dots, S$.

La matriu *preu* conté per a cada node $i = 1, \dots, dimp$ el preu de l'electricitat durant 24 hores d'un dia de la setmana.

La matriu *E* representa els preus de l'electricitat per a cada escenari $i = 1, \dots, S$ durant el nombre total d'etapes *T*, amb 24 preus per a cada etapa.

Un exemple seguint la figura 3.3:

```
pi [*] :=
1 0.402973
2 0.597027
3 0.676898
4 0.323102
5 0.0426177
6 0.957382
;

prob [*] :=
1 0.272772
2 0.130202
3 0.0254439
4 0.571583
;

preu [*,*] (tr)
:      1      2      3      4      5      6      :=
1 0.946848 0.596365 1.38207 0.0515842 0.530362 0.25781
2 2.27515 2.9972 0.428316 1.24251 0.609472 0.303486
3 0.123724 1.49645 0.657542 0.906071 0.673939 0.636112
4 1.84145 0.59326 0.34324 0.169857 0.123738 0.466507
.....
23 0.383375 1.62858 0.706802 0.621775 0.639317 0.809793
24 0.545004 0.136082 0.10489 0.0134968 0.238207 0.610413
;
```

```
E [*,*] (tr)
:      1      2      3      4      :=
1      0.946848  0.946848  0.596365  0.596365
2      2.27515   2.27515   2.9972   2.9972
3      0.123724  0.123724  1.49645  1.49645
4      1.84145   1.84145   0.59326  0.59326
.....
47     0.706802  0.621775  0.639317  0.809793
48     0.10489   0.0134968 0.238207  0.610413
;
```

Els valors d'aquests paràmetres provenen de dades reals emmagatzemades en el fitxer de dades o de dades simulades pel mateix programa a partir d'unes instruccions.

3.3.2 Codi de l'arbre d'escenaris

Un cop presentada l'estructura que s'utilitzarà per implementar l'arbre, es farà una breu descripció del codi i la seva justificació.

S, nod i dimp

A partir dels paràmetres d'entrada T i $N\{1..T\}$ calculem el nombre d'escenaris S , el nombre de nodes a cada etapa nod i el nombre de nodes total $dimp$.

Els paràmetres venen donats per les següents expressions i el seu corresponent codi

$$S = N_1 \cdot N_2 \cdot \dots \cdot N_T$$
$$nod_i = N_1 \cdot N_2 \cdot \dots \cdot N_i \quad i = 1, \dots, T$$
$$dimp = N_1 + N_1 \cdot N_2 + \dots + N_1 \cdot N_2 \cdot \dots \cdot N_T$$

```
let dimp := 0;
let S := 1;
for {i in 1..T}
{
  let S := N[i]*S;
  let nod[i] := S;
  let dimp := dimp + S;
}
```

CN i cjt

Amb el paràmetre CN volem saber quin conjunt de nodes té cada etapa. El procediment plantejat aprofita que tenim els índexs dels nodes ordenats i que amb nod sabem quants nodes hi ha a cada etapa per a poder assignar-los al conjunt al que pertanyen.

```
let CN[1] := {1..N[1]};
for {i in 2..T}
{
  let CN[i] := {last(CN[i-1]) + 1..last(CN[i-1]) + nod[i]};
}
```

El procediment per a calcular cjt és assignar, amb la informació donada per CN , a quina etapa pertany el node i .

```
for {i in 1..T}
  {
  for{j in first(CN[i])..last(CN[i])}
    {let cjt[j] := i;}
  }
```

Les funcions *first* i *last*, retornen el primer i l'últim element d'un conjunt ordenat.

A

En aquest cas aprofitem la informació donada per *CN*, *S* i *nod* per tal d'omplir la matriu *A*. El procediment a seguir és cercar els nodes que formen l'escenari al llarg de les etapes.

```
param aux1 integer;
for{t in 1..T}
{
let aux1 := 1;
for{k in 1..nod[t]}
  for{i in 1..(S / nod[t])}
  {
let A[t,aux1] := member(k,CN[t]);
let aux1 := aux1 + 1;
}
}
```

La funció *member* retorna l'element en la posició *k* d'un conjunt ordenat

ant

Calculem per a cada node a quins escenaris pertany, basant-se en el fet de tenir els nodes i els escenaris ordenats com també sabent a quina etapa pertany el node estudiat i quants nodes hi ha en aquesta.

```
param aux2 integer;
let aux2 := 0;
for{i in 1..dimp}
{
if aux2 = S then let aux2 := 0;
let ant[i] := {aux2 + 1..aux2 + (S / nod[cjt[i]])};
let aux2 := last(ant[i]);
}
```

ngrup i grup

Per a calcular *ngrup* utilitzem el nombre total d'escenaris i el nombre de nodes fills de l'última etapa per obtenir el nombre d'agrupacions d'escenaris.

```
let ngrup:=S/N[T];
```

En aquest càlcul aprofitem les dades de *CN* i *ant* per trobar els membres dels grups. Cal tenir en compte els arbres que tenen 1 etapa.

```
if T = 1 then {let grup[1]:={1..N[T]}}
else{ for{i in 1..ngrup}let grup[i]:=ant[member(i,CN[T-1])];
};
```

Probabilitats i preus

Quan utilitzem les dades de preus d'electricitat reals i les probabilitats estan fixades, hem de cridar al fitxer de dades on estan emmagatzemades amb la següent instrucció:

```
data arbre.dat;
```

Aquesta comanda assignaria directament valors als paràmetres *prob* i *E* en aquest cas.

En el cas d'utilitzar dades simulades, aquestes es simulen amb generadors de nombres aleatoris de distribució normal i uniforme.

Per a crear probabilitats simulades no equiprobables per a cada node generem *k* nombres aleatoris, distribuïts segons una llei Uniforme[0,1], que corresponen al nombre de nodes fills que surten d'aquest node segons l'etapa on estigui situat.

Per assegurar que les probabilitats compleixin la restricció de sumar 1, fem el següent procediment:

Generem $pr = (pr_1, \dots, pr_k)$;

Calculem $\sum_{i=1}^k pr_i$;

Assignem $pr = \frac{(pr_1, \dots, pr_k)}{\sum_{i=1}^k pr_i}$;

Després assignem aquestes probabilitats a la posició del vector *pi* que li pertoca.

```
param suma;
param pr {1..dimp};
param aux3 integer;
param aux4 integer;

let aux3 := 0;
for{t in 1..T}
  for{j in 1..(nod[t] / N[t])}
    {
      for{k in 1..N[t]}
        {
          let pr[k] := Uniform01();
        }
      let suma := sum{i in 1..N[t]}(pr[i]);
      for{i in 1..N[t]}
        {
          let pr[i] := (pr[i] / suma);
        }
      let aux4 := 1;
      for{m in aux3 + 1..aux3 + N[t]}
        {
          let pi[m] := pr[aux4];
          let aux4 := aux4 + 1;
        }
      let aux3 := aux3 + N[t];
    }
}
```


La funció `Uniform01()` genera un nombre aleatori que es distribueix segons una llei Uniforme[0,1].

En el cas que l'arbre fos equiprobable tenim el següent procediment:

```
param aux3 integer;
param aux4 integer;
let aux3 := 0;
for{t in 1..T}
  for{j in 1..(nod[t] / N[t])}
    {
      let aux4 := 1;
      for{m in aux3 + 1..aux3 + N[t]}
        {
          let pi[m] := 1/N[t];
          let aux4 := aux4 + 1;
        }
      let aux3 := aux3 + N[t];
    }
}
```

El següent pas és calcular les probabilitats dels escenaris, a partir de la informació de la matriu A .

$$prob_i = \prod_{k \in K} pi_k \quad \text{on } i = 1, \dots, S; K = \{\text{conjunt de nodes que pertanyen a l'escenari } i\}$$

```
for{i in 1..S}
{
  let prob[i] := 1;
  for{t in 1..T}
    {
      let prob[i] := pi[A[t,i]] * prob[i];
    }
}
```

Pel que fa els preus es generen aleatòriament a partir de la funció `Normal01()` o `Normal(mu, sigma)`. I els escenaris es van construir a partir dels preus i de la matriu de predecessors A :

```
for{i in 1..dimp, j in 1..24}
  let preu[i,j] := Normal(5,1);
param aux5 integer;
for{i in 1..S}
{
  let aux5 := 1;
  for{j in 1..T}{
    for{k in 1..24}
      {
        let E[i,aux5] := preu[A[j,i],k];
        let aux5 := aux5 + 1;
      }
  }
}
```

3.4 Implementació dels algorismes

El principal objectiu és aplicar un mètode de reducció per a poder obtenir a partir d'un arbre original un arbre amb un nombre més petit d'escenaris.

En aquest cas, els mètodes implementats són el *Simultaneous Backward Reduction* i *Fast Forward Selection*, els quals han estat exposats en l'apartat 3.1.3.

3.4.1 Elements de la implementació dels algorismes

Per aquesta implementació partim amb els elements presentats en l'apartat 3.2, i el paràmetre d'entrada, assignat directament del fitxer de dades:

```
param NJ integer;
```

El paràmetre NJ correspon a la cardinalitat del conjunt J , és a dir, de quants escenaris han de ser eliminats.

Per a poder calcular les distàncies entre els parells d'escenaris:

```
param ct {1..S,1..S};  
param c {1..S,1..S};
```

La matriu ct guarda el valor de la distància entre els parells d'escenaris $i=1,\dots,S$ i $j=1,\dots,S$.

La matriu c també guarda el valors de les distàncies, però aquestes s'actualitzen a cada passa de l'algorisme.

Necessitem obtenir una llista dels escenaris que eliminen i els que preservem, aquesta informació s'obté amb els conjunts:

```
set J {1..1} ordered;  
set L {1..1} ordered;
```

El conjunt indexat i ordenat J indica l'índex dels escenaris que han estat eliminats.

El conjunt indexat i ordenat L indica l'índex dels escenaris que han estat preservats.

Cal notar que s'ha representat els conjunts com a indexats, per poder iterar sobre ells.

Per representar el paràmetre que guarda les distàncies de probabilitat i el seu *argmin*:

```
param z {1..S};  
set z_argmin {1..1} ordered;
```

El vector z representa el valor del producte de la mínima distància a l'escenari $l=1,\dots,S$ i la probabilitat de l'escenari l .

El conjunt indexat i ordenat z_argmin indica quin és l'índex del valor mínim del vector z .

Per a poder assignar les noves probabilitats dels escenaris preservats utilitzem:

```
set minim{1..1} ordered;
```

El conjunt indexat i ordenat *minim* indica quin és l'escenari $i \in L$ que és més proper a l'escenari $j \in J$.

3.4.2 Codi de Simultaneous Backward Reduction

La idea principal d'aquest algorisme es basa en l'eliminació òptima d'un escenari, en cada iteració, a partir d'un criteri de distància de probabilitat. Aquesta procés es repeteix fins a obtenir un nombre fixat $S - s$ d'escenaris eliminats.

A partir dels elements presentats en l'apartat anterior, duem a terme la implementació de l'algorisme dividit en les següents etapes o passos :

Pas 0

Duem a terme el càlcul de les distàncies entre parells d'escenaris.

$$c_{kj} := c_T(\xi^k, \xi^j), k, j = 1, \dots, S.$$

```
for{i in 1..S, j in 1..S}
  let ct[i, j] := sum{k in 1..24*T} (abs(E[i, k] - E[j, k]));
```

Les dades dels preus del escenaris estan contingudes en la matriu E i el resultat de les distàncies es guarden en la matriu ct .

Pas 1

Calculem:

$$c_{ll}^{[1]} := \min_{j \neq l} c_{lj}, l = 1, \dots, S$$

$$z_l^{[1]} := p_l c_{ll}^{[1]}, l = 1, \dots, S.$$

$$\text{Tria } l_1 \in \arg \min_{l \in \{1, \dots, S\}} z_l^{[1]}.$$

$$\text{Conjunt } J^{[1]} := \{l_1\}.$$

```
for{l in 1..S}
{
  let c[l, l] := min {j in 1..S diff{l}} ct[l, j];
}
```

```
for{l in 1..S}
{
  let z[l] := prob[l] * c[l, l];
}
```

```
let z_argmin[1] := {i in 1..S: z[i] = min {j in 1..S} z[j]};  
let J[1] := {first(z_argmin[1])};  
let L[1] := 1..S diff {first(z_argmin[1])};
```

En aquest pas, la matriu c emmagatzema el valor de la distància mínima de cada escenari.

El vector z indica el valor del producte d'aquesta distància guardada en c amb la probabilitat de l'escenari.

Llavors escollim l'índex del valor mínim del vector z , i l'afegim a la llista d'escenaris eliminats J i l'eliminem del conjunt L .

Pas i

Calculem:

$$c_{kl}^{[i]} := \min_{j \in J^{[i-1]} \cup \{l\}} c_{kj} \text{ per a } l \notin J^{[i-1]}, k \in J^{[i-1]} \cup \{l\}$$
$$z_l^{[i]} := \sum_{k \in J^{[i-1]} \cup \{l\}} p_k c_{kl}^{[i]}, l \notin J^{[i-1]}.$$

$$\text{Tria } l_i \in \arg \min_{l \notin J^{[i-1]}} z_l^{[i]}.$$

$$\text{Conjunt } J^{[i]} := J^{[i-1]} \cup \{l_i\}.$$

```
for{n in 2..NJ}  
{  
  for{l in L[1]}  
  {  
    for{k in {l} union J[1]}  
    {  
      let c[k,l] := min {j in L[1] diff {l}} ct[k,j];  
    }  
  }  
  for{l in L[1]}  
  {  
    let z[l] := sum {k in {l} union J[1]}(prob[k] * c[k,l]);  
  }  
  let z_argmin[1] := {i in L[1]: z[i] = min {j in L[1]} z[j]};  
  let J[1] := J[1] union {first(z_argmin[1])};  
  let L[1] := L[1] diff {first(z_argmin[1])};  
}
```

En aquest pas repetim el procediment del pas 1, tenint en compte els escenaris eliminats en el càlcul de la matriu c i el vector z .

El procés es repeteix fins que aconseguim un nombre NJ d'escenaris eliminats.

Pas S-s+1

Tenim el conjunt:

$$J := J^{[S-s]}$$

Hem de calcular les probabilitats del escenaris preservats amb la *regla de redistribució òptima* (3.2).

```
for {i in J[1]}
{
let mínim[1] := {j in L[1]: ct[i,j] = min {k in L[1]} ct[i,k]};
let prob[first(mínim[1])] := prob[first(mínim[1])] + prob[i];
let prob[i] := 0;
}
```

En aquest pas , per a cada escenari eliminat busquem l'escenari preservat més proper respecte les dades de la matriu *ct* i li afegim la probabilitat de l'escenari eliminat. A tots els escenaris eliminats els assignem probabilitat 0.

3.4.3 Codi de Fast Forward Selection

Aquest mètode es basa en la selecció òptima d'un escenari en cada iteració a partir d'un criteri de distància de probabilitat. Aquesta procés es repeteix fins a obtenir un nombre fixat *s* d'escenaris preservats.

La implementació d'aquest algorisme utilitza els elements descrits en l'apartats 3.4.1 i consta dels següents passos:

Pas 0

Duem a terme el càlcul de les distàncies entre parells d'escenaris.

$$c_{ku}^{[i]} := c_T(\xi^k, \xi^u), k, u = 1, \dots, S.$$

```
for{i in 1..S}
for{j in 1..S}
{
let ct[i,j] := sum{k in 1..24*T}(abs(E[i,k] - E[j,k]));
let c[i,j] := ct[i,j];
}
```

Les dades dels preus del escenaris estan contingudes en la matriu *E*, el resultat de les distàncies es guarden en la matriu *ct* i es copien en la matriu *c*, que s'anirà actualitzant a cada iteració de l'algorisme.

Pas 1

Calculem:

$$z_u^{[1]} := \sum_{\substack{k=1 \\ k \neq u}} p_k c_{ku}^{[1]}, \quad u = 1, \dots, S.$$

$$\text{Tria } u_1 \in \arg \min_{u \in \{1, \dots, S\}} z_u^{[1]}.$$

$$\text{Conjunt } J^{[1]} := \{1, \dots, S\} \setminus \{u_1\}.$$

```
for{u in 1..S}
{
  let z[u] := sum {k in 1..S diff {u}}(prob[k] * c[k,u]);
}
let z_argmin[1] := {i in 1..S: z[i] = min {j in 1..S} z[j]};
let J[1] := 1..S diff {first(z_argmin[1])};
```

El vector z indica el valor del producte de les distàncies de c amb la probabilitat de l'escenari.

Llavors escollim l'índex del valor mínim del vector z , i assignem a J el conjunt $\{1, \dots, S\}$ diferència $first(z_argmin)$.

Pas i

Calculem:

$$c_{ku}^{[i]} := \min \{c_{ku}^{[i-1]}, c_{ku_{i-2}}^{[i-1]}\}, \quad k, u \in J^{[i-1]}$$

$$z_u^{[i]} := \sum_{k \in J^{[i-1]} \setminus \{u\}} p_k c_{ku}^{[i]}, \quad u \in J^{[i-1]}.$$

$$\text{Tria } u_i \in \arg \min_{u \in J^{[i-1]}} z_u^{[i]}.$$

$$\text{Conjunt } J^{[i]} := J^{[i-1]} \setminus \{u_i\}.$$

```
for{n in 2..(S - NJ)}
{
  for{k in J[1]}
  {
    for{u in J[1]}
    {
      let c[k,u] := min( c[k,u], c[k,first(z_argmin[1])] );
    }
  }
}
for{u in J[1]}
{
  let z[u] := sum{k in J[1] diff {u}}(q[k] * c[k,u]);
}
let z_argmin[1] := {i in J[1]: z[i] = min {j in J[1]} z[j]};
let J[1] := J[1] diff {first(z_argmin[1])};
}
```

En aquest pas actualitzem els valors de la matriu c , comparant la distància actualitzada amb el valor de la distància a l'escenari preservat en la iteració anterior.

Calculem novament el vector z , seleccionem l'índex del valor mínim i actualitzem el conjunt J .

Aquest procés es repeteix fins que aconseguim un nombre $S - NJ$ d'escenaris preservats.

Pas $s+1$

Tenim el conjunt:

$$J := J^{[S-s]}$$

Hem de calcular les probabilitats del escenaris preservats amb la *regla de redistribució òptima* (3.2).

```
for {i in J[1]}
{
let minim[1] := {j in L: ct[i,j] = min {k in L} ct[i,k]};
let prob[first(minim[1])] := prob[first(minim[1])] + prob[i];
let prob[i] := 0;
}
```

Aquesta regla marca que per a cada escenari eliminat busquem l'escenari preservat més proper respecte les dades de la matriu ct i li afegim la probabilitat de l'escenari eliminat. A tots els escenaris eliminats els assignem probabilitat 0.

3.5 Resultats

Un cop aplicat qualsevol dels dos mètodes sobre l'arbre d'escenaris, s'obté un subconjunt del conjunt d'escenaris originals i una nova distribució de probabilitat per als escenaris preservats $s = S - \#J$.

3.5.1 Variables de sortida

Per tal de poder representar el nou arbre existent i tenir les dades necessàries per al model d'optimització s'han creat els següents elements

```
param nBScen integer;  
set BScen{1..nBScen}within {1..S} circular;
```

El paràmetre $nBScen$ indica en quants grups s'agrupen els escenaris.

El conjunt $BScen$ correspon al conjunt ordenat d'escenaris $i = 1, \dots, s$ que pertanyen al grup d'escenaris $j = 1, \dots, nBScen$.

Aquestes dues variables serviran per poder formular les constriccions de no anticipativitat en models de 2 etapes. Per representar els preus i la nova distribució de probabilitat:

```
param q {1..S-NJ};  
param Sc {1..S-NJ, 1..24*T};
```

El vector q indica la probabilitat de cada escenari preservats $i = 1, \dots, s$.

La matriu Sc representa els preus de l'electricitat dels escenaris preservats $i = 1, \dots, s$ durant el nombre total d'etapes T , amb 24 preus per a cada etapa.

Per exemple, els paràmetres presentats podrien obtenir els següents valors en el cas de la figura 3.2 i amb el nombre d'escenaris a eliminar $NJ = 2$.

```
nBScen = 2;  
  
set Bscen[1] := 1;  
set Bscen[2] := 2;  
  
q [*] :=  
1 0.400896  
2 0.599104  
;  
  
Sc[*,*] (tr)  
:      1      2      :=  
1  4.63561  3.59456  
2  3.96165  4.26127  
.....  
47  5.69924  3.97463  
48  3.64486  6.05417  
;
```


3.5.2 Codi de les variables de sortida

Presentem el codi de la implementació de les variables presentades en l'apartat 3.5.1

Primer de tot, actualitzem la variable *grup*, excloent els escenaris eliminats, per tal d'obtenir informació sobre la nova composició dels escenaris:

```
for{i in 1..ngrup}
{
let grup[i]:=grup[i] diff J[1];
}
```

nBScen

Per tal d'aconseguir el nombre de grups d'escenaris presents en el nou arbre, utilitzem *nBScen* com a comptador no tenint en compte el conjunt de grups que siguin buits.

```
let nBScen:=0;
for{n in 1..ngrup}{
if(card(grup[n])!=0) then{let nBScen:=nBScen+1};
}
```

La funció *card* retorna el cardinal d'un conjunt.

BScen

En aquest cas hem de reindexar els grups d'escenaris que tenen escenaris preservats, és a dir que no tinguin cardinal nul.

```
param aux6 integer;
param aux7 integer;
let aux6:=1;
let aux7:=1;
for{n in 1..ngrup}{
if(card(grup[n])!=0) then{
for{i in first(grup[n])..last(grup[n])}
{
let BScen[aux7]:={aux6};
let aux6:=aux6+1;
}
let aux7:=aux7+1;
}
}
```

q i prob

El procediment es basa en reordenar els índexs de les probabilitats i els preus dels escenaris preservats.

```
for{i in 1..S-NJ}{
let q[i]:=prob[member(i,L[1])];}
for{i in 1..S-NJ}{
for{j in 1..24*T}
let Sc[i,j]:=E[member(i,L[1]),j];}
```

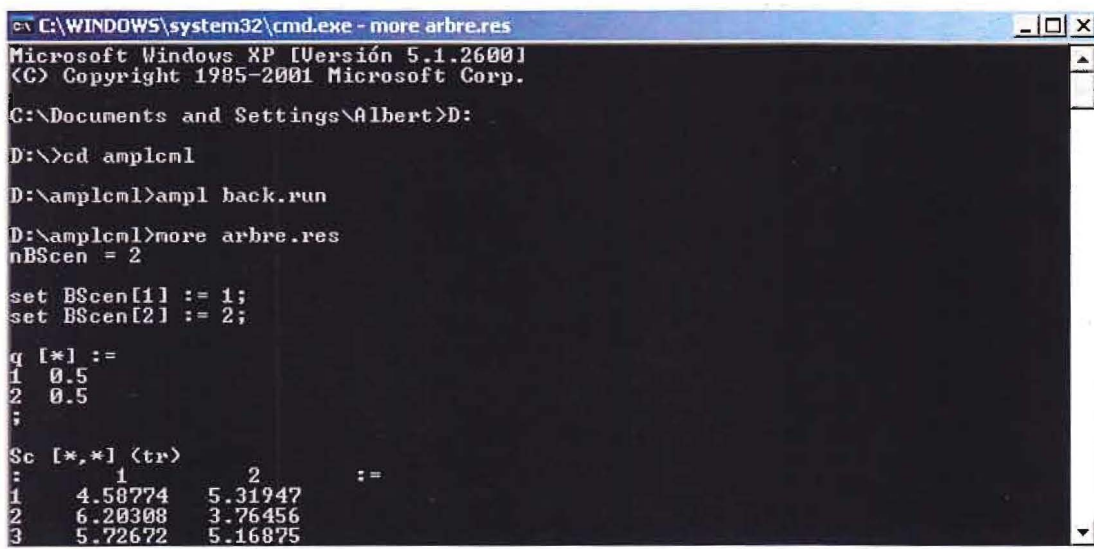
3.6 Execució

Per poder dur a terme l'execució dels algorismes necessitarem per una banda el fitxer de comandes anteriorment descrit i el fitxer de dades, el qual l'usuari ha de manipular per a obtenir el problema desitjat.

Cal recordar que aquest fitxer conté els següents elements:

- Paràmetre T amb el nombre d'etapes.
- Paràmetre N correspon al nombre de nodes fills que hi ha en cada etapa $t = 1, \dots, T$.
- Paràmetre NJ , nombre d'escenaris eliminats.
- Vector $prob$, distribució de probabilitat dels escenaris originals, opcional.
- Matriu E , preus dels escenaris originals, opcional.

Llavors, executant el programa amb AMPL des de la pantalla de comandes (figura 3.4) obtenim un seguit d'elements que són guardats en un fitxer de resultats.



```
C:\WINDOWS\system32\cmd.exe - more arbre.res
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Albert>D:
D:\>cd amplcml
D:\amplcml>ampl back.run
D:\amplcml>more arbre.res
nBScen = 2

set BScen[1] := 1;
set BScen[2] := 2;

q [*] :=
1 0.5
2 0.5
;

Sc [*,*] (tr)
:      1      2      :=
1 4.58774 5.31947
2 6.20308 3.76456
3 5.72672 5.16875
```

Figura 3.4: Exemple d'execució en AMPL de Simultaneous Backward Reduction en Windows XP

Els elements que retorna:

- Paràmetre $nBScen$, nombre de grups d'escenaris preservats
- Conjunt $BScen$, grups d'escenaris que contenen els escenaris preservats.
- Vector q , nova distribució de probabilitat dels escenaris preservats.
- Matriu Sc , preus dels escenaris preservats.

Aquestes dades obtingudes s'utilitzaran per a poder resoldre el problema d'optimització de l'oferta al mercat elèctric diari.

3.6.1 Exemple

Per il·lustrar els resultats obtinguts en l'execució proposem el següent cas senzill, en el fitxer de dades tenim els següents valors:

```
param T := 2;
param N:= 1 2 2 2;
param NJ:= 2;
```

En la figura 3.5 podem observar l'arbre d'escenaris simulat que s'ha generat amb aquests paràmetres, i també podem les variables que defineixen i contenen la informació dels escenaris:

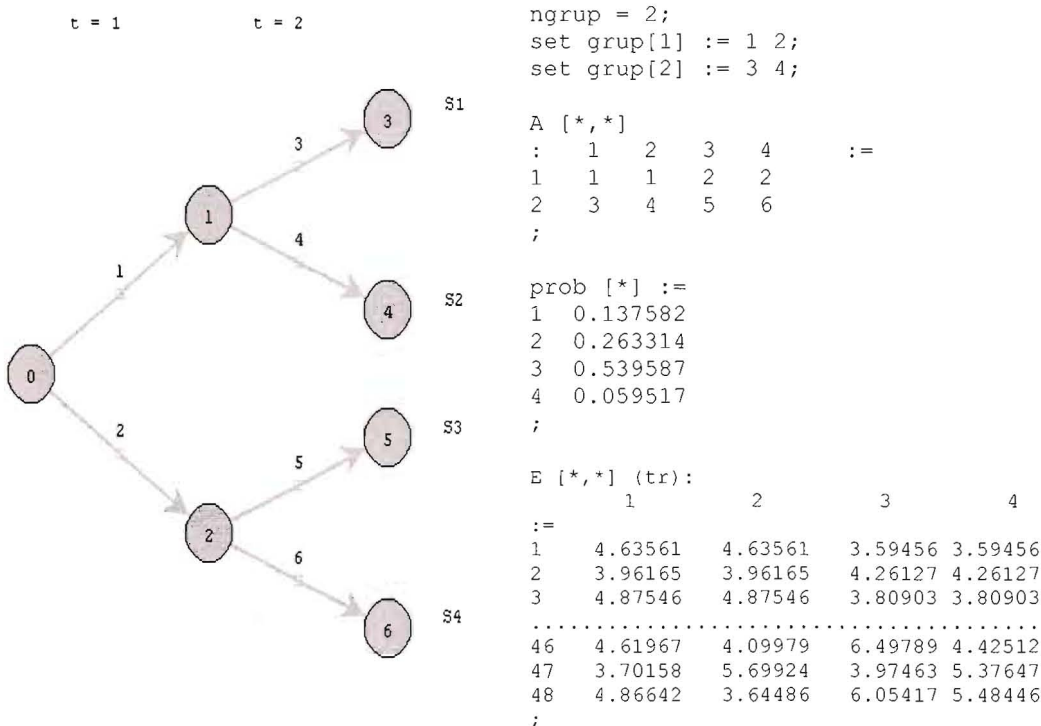


Figura 3.5: Arbre d'escenaris inicial

En la figura 3.6 es presenta l'arbre reduït després de l'execució dels algorismes. Podem veure marcat en verd els escenaris preservats, que no pertanyen al conjunt J . Obtenim la matriu de predecessors A i la matriu dels preus dels escenaris E reduïdes als escenaris preservats. Com també obtenim la nova distribució de probabilitat del escenaris preservats.

```
set J[1] := 4 1;
nBScen = 2;
set BScen[1] := 1;
set BScen[2] := 2;
```

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

```

A[i,j]
: 1 2 :=
1 1 2
2 3 4
;

q[*] :=
1 0.400896
2 0.599104
;

Sc [*,*] (tr)
: 1 2 :=
1 4.63561 3.59456
2 3.96165 4.26127
.....
47 5.69924 3.9746
48 3.64486 6.05417
;

```

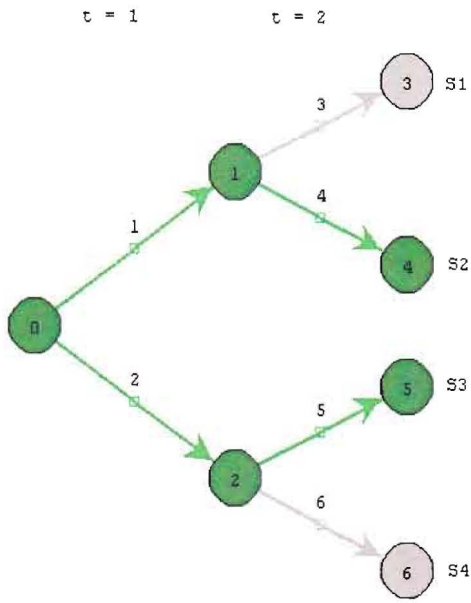


Figura 3.6: Arbre d'escenaris reduït

4- Proves computacionals

4.1 Proves amb arbres d'escenaris

Un cop dissenyats i implementats els dos algorismes de reducció d'escenaris es va procedir a realitzar un seguit de tests computacionals. L'objectiu d'aquestes proves era comprovar el correcte funcionament de la implementació i millorar la seva eficiència.

Cal tenir en compte que s'han implementant algorismes que utilitzen mètodes heurístics, és a dir, que proporcionen solucions, però no hi ha garantia de que siguin òptimes.

Els casos que es presenten seguidament van ser realitzar en un ordinador Intel® Pentium® amb un processador de 3.40 Ghz i 512 MB de RAM, en la Facultat de Matemàtiques i Estadística.

4.1.1 Cas 1

En aquest primer cas es va considerar construir un arbre simulat per poder obtenir les primeres conclusions sobre la bondat de la implementació i poder comparar el comportament de les dues metodologies.

Es presenta un arbre d'escenaris amb probabilitats i preus simulats i amb paràmetres $T = 2$ i $N = (10,10)$. Amb aquesta estructura s'obté un nombre $S = 100$ d'escenaris, com es pot observar en la figura 4.1:

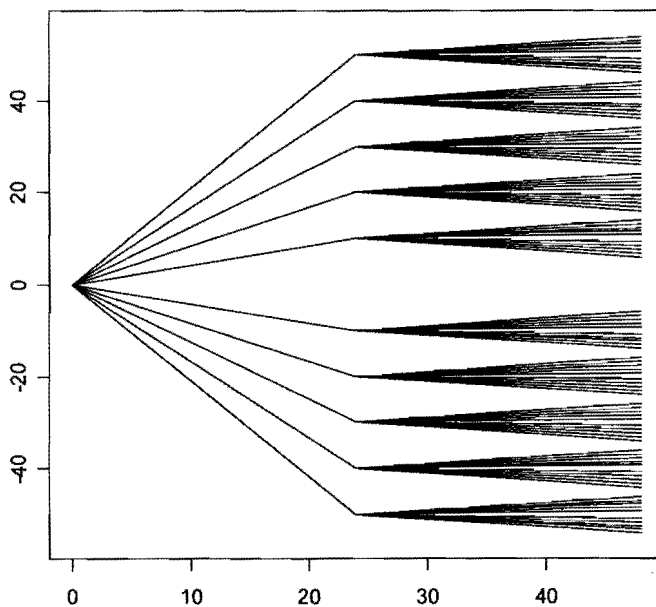


Figura 4.1: Representació Arbre 100 escenaris

Es va proposar $s = 20$ i $s = 10$ com a nombres d'escenaris preservats, aconseguint els següents resultats:

Simultaneous Backward Reduction

s = 20

```
set L :=  
3    10   23   28   30   36   39   56   68   90  
6    18   26   29   35   38   55   63   77   100;
```

s = 10

```
set L := 6 28 30 35 38 55 68 77 90 100;
```

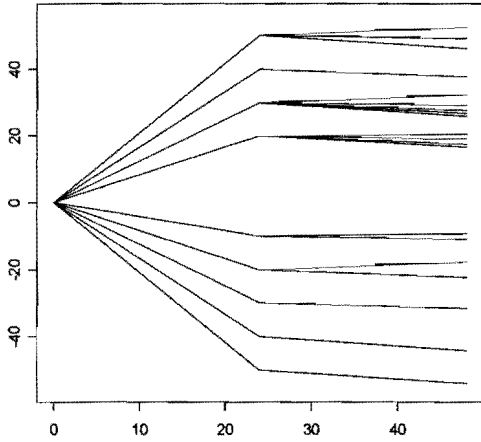


Figura 4.2: Arbre reduït 20 escenaris

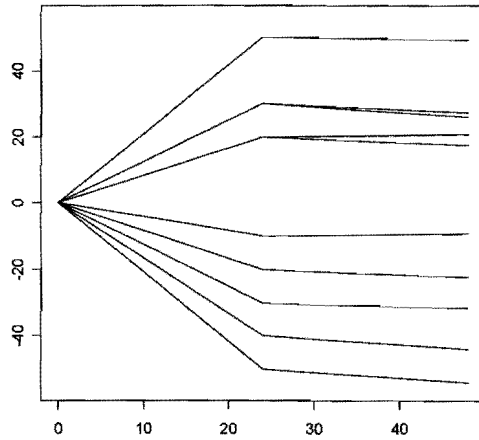


Figura 4.3: Arbre reduït 10 escenaris

Fast Forward Selection

s = 20

```
set L := 3 7 10 18 23 24 28 29 30 35 36 38 39 55 56 63 68 77 90 99;
```

s = 10

```
set L := 7 24 35 38 39 55 68 77 90 99;
```

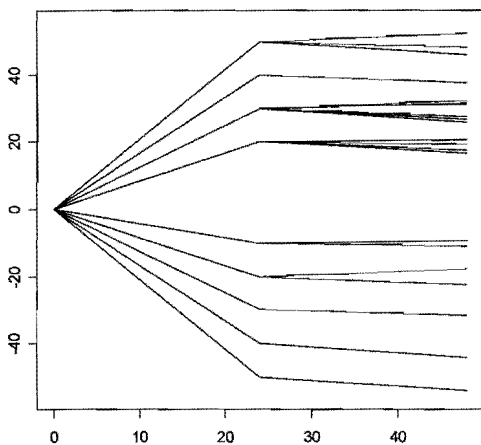


Figura 4.4: Arbre reduït 20 escenaris

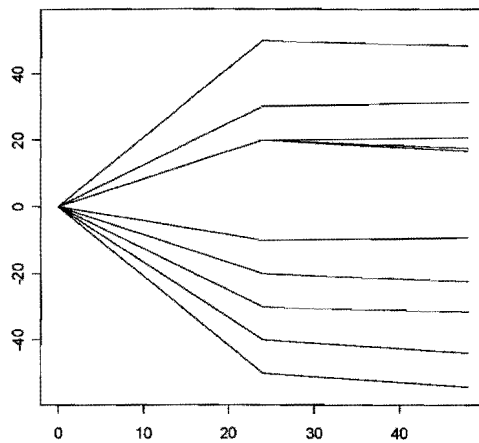


Figura 4.5: Arbre reduït 10 escenaris

Es pot observar que els dos mètodes arriben a solucions diferents del mateix arbre, encara que són força semblants. Alhora, es pot comprovar que, en els dos casos, la llista de 20 escenaris engloba la de 10, aquest fet mostra que no hi ha cap comportament irregular de l'algorisme, que ha repetit les mateixes passes obtenint resultats idèntics.

Temps d'execució

Per poder tenir un altre element de judici per comparar els algorismes, s'ha realitzat un estudi dels temps d'execució segons els escenaris conservats (s) del total d'escenaris (S):

s/S	TEMPS (s)	
	Backward	Forward
20/100	9	92
10/100	10	84
5/100	11	66
3/100	11	43
2/100	11	32
1/100	12	2

Taula 4.1: Temps d'execució algorismes

En la taula 4.1 es pot veure com el temps de Backward és clarament inferior al de Forward, ja que aquest ha de realitzar un nombre de passes més gran per a poder aconseguir una solució. Això pot ser degut a que ha de realitzar, en cada iteració, una cerca per pràcticament totes les caselles de la matriu de distàncies i realitzar comparacions, cosa molt poc eficient.

Es pot apreciar que a mesura que decreix *s* el temps de Forward disminueix, ja que al seleccionar un nombre menor d'escenaris realitza menys iteracions, de fet només supera el temps de Backward quan hi ha una forta reducció.

Pel cas de Backward, al augmentar el nombre d'escenaris a conservar ha de realitzar més iteracions ja que ha d'eliminar un nombre més gran d'escenaris, fet pel qual augmenta el temps d'execució.

Cal notar que totes les execucions han acabat satisfactòriament, ja que han convergit cap a una solució.

4.1.2 Cas 2

En aquest cas es va utilitzar un arbre fan, el qual tenia 1 etapa i escenaris equiprobables. Per a poder validar que els algorismes obtenien els resultats desitjats, es va procedir a construir el fan a partir de replicar i modificar escenaris reals, les dades es poden trobar en el CD adjunt al PFC.

L'objectiu era que aquests escenaris reals fossin els preservats, fet que mostraria el correcte funcionament de la implementació, ja que eliminaria els escenaris redundants i conservariem la informació vàlida.

Es van utilitzar un nombre $S = 154$ d'escenaris, i amb el paràmetre $s = 20$ i $s = 10$.

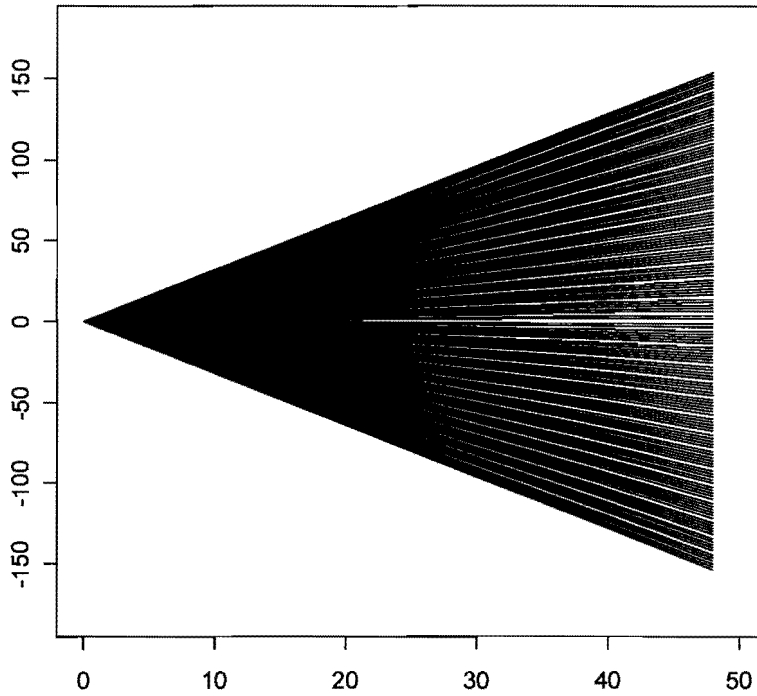


Figura 4.6: Fan de 154 escenaris

Simultaneous Backward Reduction

s = 20

```
set L :=  
120 122 124 126 128 130 132 134 136 138  
121 123 125 127 129 131 133 135 137 139;
```

s = 10

```
set L := 120 122 123 125 128 129 130 131 133 137;
```

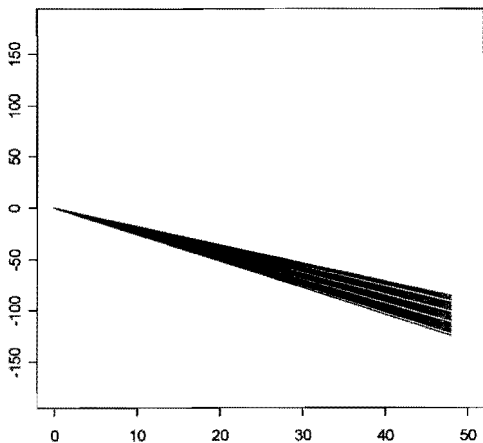


Figura 4.7: Fan reduït 20 escenaris

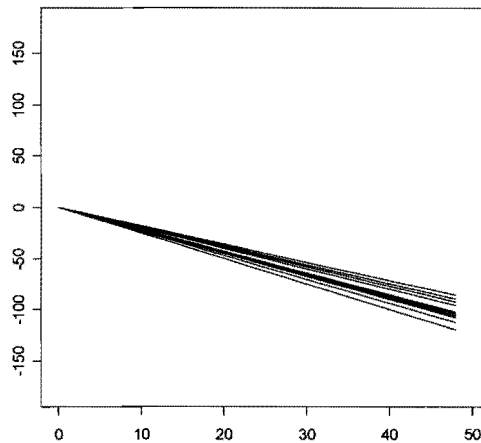


Figura 4.8: Fan reduït 10 escenaris

Fast Forward Selection

s = 20

```
set L := 1 2 3 4 5 7 8 9 10 11 12 13 14 15 16 17 18 19 20 42;
```

s = 10

```
set L := 1 3 4 5 10 11 12 13 19 42;
```

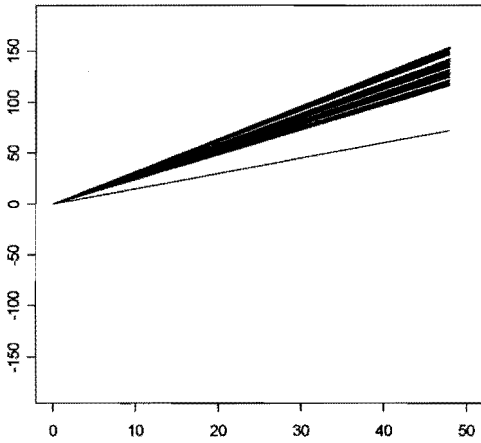


Figura 4.9: Fan reduït 20 escenaris

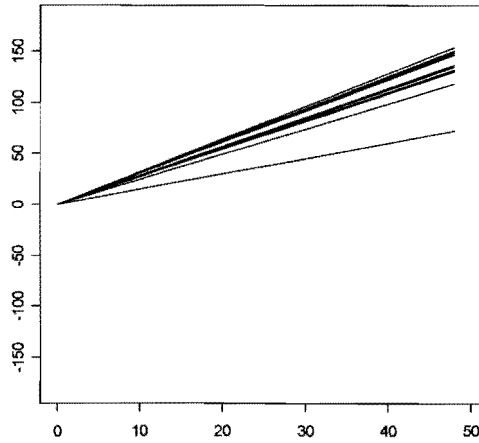


Figura 4.10: Fan reduït 10 escenaris

En l'execució de l'algorisme Backward s'aconsegueixen resultats satisfactoris, ja que els escenaris obtinguts corresponen a 20 escenaris que tenen exactament el mateix valor que els 20 primers escenaris originals. El subconjunt de 10 està inclòs en aquest.

En canvi, pel que fa a Forward obtenim majoritàriament escenaris originals menys l'escenari 42 que correspon a un escenari modificat. Aquest error és per culpa de la metodologia que utilitza l'algorisme, ja que aquest escenari conté la suma de distàncies mínima en la primera iteració, per tant, ha de ser seleccionat com a escenari preservat pel criteri aplicat en Forward.

Temps d'execució

Obtenim la següent taula 4.2 corresponent als temps d'execució:

s/S	TEMPS (s)	
	Backward	Forward
20/154	19	392
10/154	25	303

Taula 4.2: Temps d'execució algorismes

Es pot comprovar novament que el temps de Backward és clarament inferior al de Forward.

Amb aquestes dades es podria acceptar que l'algorisme de Simultaneous Backward Reduction obté millors resultats, tant en correcció com en eficiència.

4.1.3 Cas 3

En aquest tercer cas es va construir un arbre d'escenaris de dues etapes amb preus reals del mercat elèctric i escenaris equiprobables, la totalitat d'aquestes dades es poden trobar en el CD adjunt al PFC.

Es va considerar com a paràmetres de l'arbre $T = 2$ i $N = (30,10)$. El que generava un nombre $S = 300$ d'escenaris, i es va proposar reduccions a $s = 200$, $s = 100$, $s = 50$, $s = 25$ i $s = 10$.

Es volia comprovar en quina mesura els arbres reduïts mantenen l'estructura dels preus respecte l'arbre original, es va dur a terme l'estudi de la mitjana del preu de cada hora $i = 1, \dots, 48$, a partir de la mostra d'escenaris que composarien l'arbre en $s \in \{200, 100, 50, 25, 10\}$.

En la figura 4.11 s'estudia el comportament del preu de l'electricitat i la seva variabilitat ($\bar{x}_i^{300} + \sigma_i^{300}$) segons l'hora durant dos dies. Es pot observar l'efecte de les hores vall i hores puntes, i un marcat patró d'estacionalitat, es pot considerar que hi ha una variabilitat més alta en les últimes hores del segon dia.

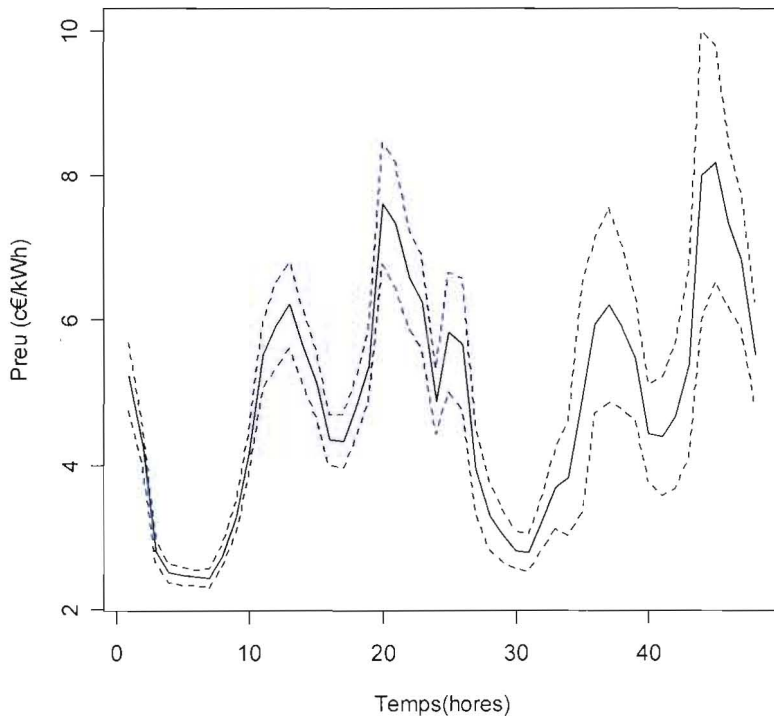


Figura 4.11: Mitjanes més desv. estàndards horàries arbre 300 escenaris

Simultaneous Backward Reduction

En aquest cas després de realitzar les reduccions mitjançant l'algorisme de reducció Backward amb $s = 200, s = 100, s = 50, s = 25$ i $s = 10$, obtenint el següents resultats:

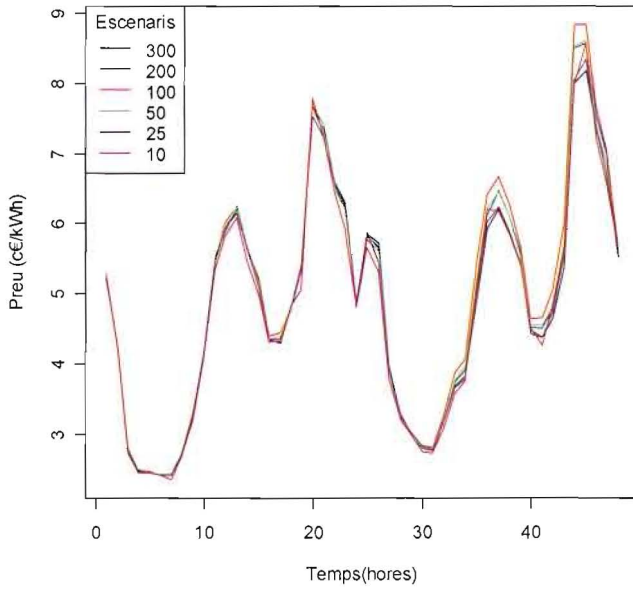


Figura 4.12: Comparació mitjanes horàries arbre escenaris reduïts

En la figura 4.12 s'observa com no hi ha diferències significatives entre l'arbre inicial i les seves successives reduccions, encara que cap a les últimes observacions els arbres reduïts ajusten amb més error.

Per a poder quantificar l'error existent entre els moments dels nous arbres d'escenaris reduïts i els moments de l'arbre l'original es va calcular a partir del criteri:

$$EQM^s = \sum_{i=1}^{48} \frac{(\bar{x}_i^s - \bar{x}_i^{300})^2}{\bar{x}_i^{300}} \text{ amb } s \in \{200, 100, 50, 25, 10\} \tag{4.1}$$

Escenaris conservats (s)	200	100	50	25	10
EQM^s	0,139903	0,517000	0,173095	0,0390932	0,184798

Taula 4.3: Error Quadràtic Mig de la mitjana

Els resultats de la taula 4.3 són antiintuïtius, ja que no segueixen el patró esperat d'augmentar l'error a mesura que anem disminuint la mostra d'escenaris, aquest fet pot ser degut a que existeixen subconjunts d'escenaris que ajusten millor les propietats de l'arbre original, però l'algorisme al variar a cada iteració el subconjunt d'escenaris pot estar variant la solució òptima.

Fast Forward Selection

Repetint el mateix procediment de reduir l'arbre original a un arbre amb $s = 200$, $s = 100$, $s = 50$, $s = 25$ i $s = 10$, s'aconsegueix la següent informació:

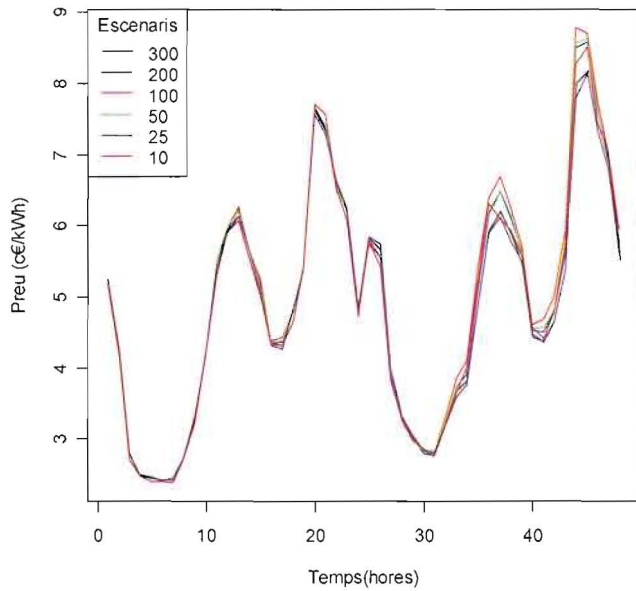


Figura 4.13: Comparació mitjanes horàries arbre escenaris reduïts

En la figura 4.13 es pot observar que amb l'algorisme *Fast Forward Selection* s'obtenen uns arbres reduïts molts semblants a l'arbre original, no hi ha cap diferència de comportament destacable, encara que tornen haver petits desajustos cap a les hores finals.

Usant el criteri (4.1) s'aconsegueixen els resultats de l'error (Taula 4.4) de la mitjana a mesura que es redueix l'arbre original. Es torna a trobar el mateix tipus de resultats antiintuïtius, a causa de la selecció dels subconjunts d'escenaris.

Escenaris conservats (s)	200	100	50	25	10
EQM^s	0,141233	0,471857	0,210741	0,0325771	0,219604

Taula 4.4: Error Quadràtic Mig de la mitjana i la desv. típica

Amb aquests resultats, es podria afirmar que els algorismes de reducció d'escenaris *Simultaneous Backward Reduction* i *Fast Forward Selection* proporcionen aproximacions raonablement bones de l'arbre original, encara que probablement alguna reducció no sigui l'òptima. Tot i així, són capaços d'aconseguir arbre d'escenaris reduïts que no fan variar significativament els moments de la variable aleatòria preu.

4.2 Aplicació al problema d'oferta òptima al mercat diari d'energia elèctrica

A continuació es presenten els resultats del problema d'optimització descrit a l'apartat 2.2.3 corresponents als arbres generats al cas 3. Aquests resultats es mostren per il·lustrar la utilitat final dels arbres obtinguts mitjançant les tècniques presentades. S'ha resolt el model d'optimització amb arbres de 10, 50, 100 i 200 escenaris, a les figures 4.14 - 4.17 es representa el valor a l'òptim de les variables de decisió del problema d'optimització per a la tèrmica 2.

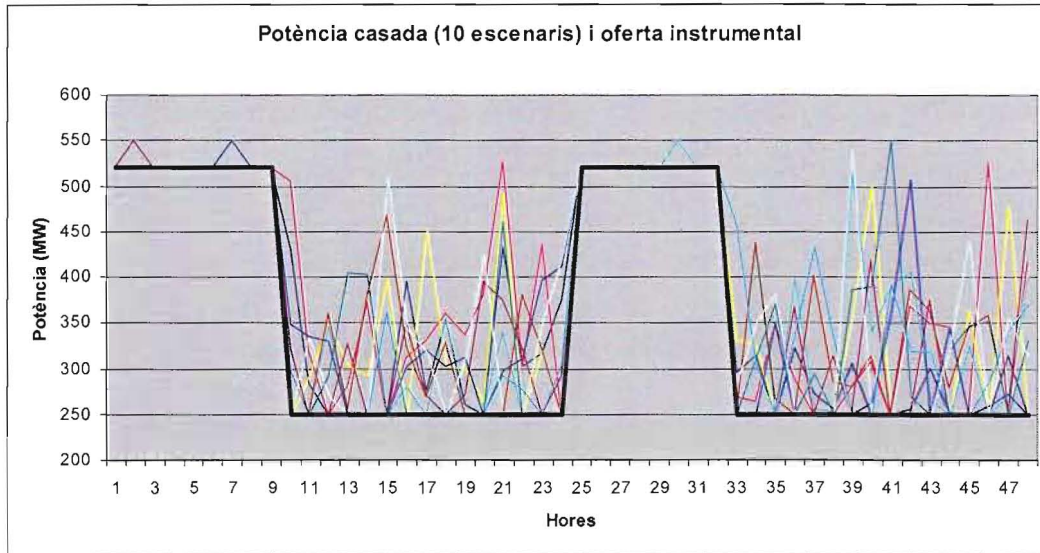


Figura 4.14: Potència cassada per 10 escenaris i oferta instrumental tèrmica 2

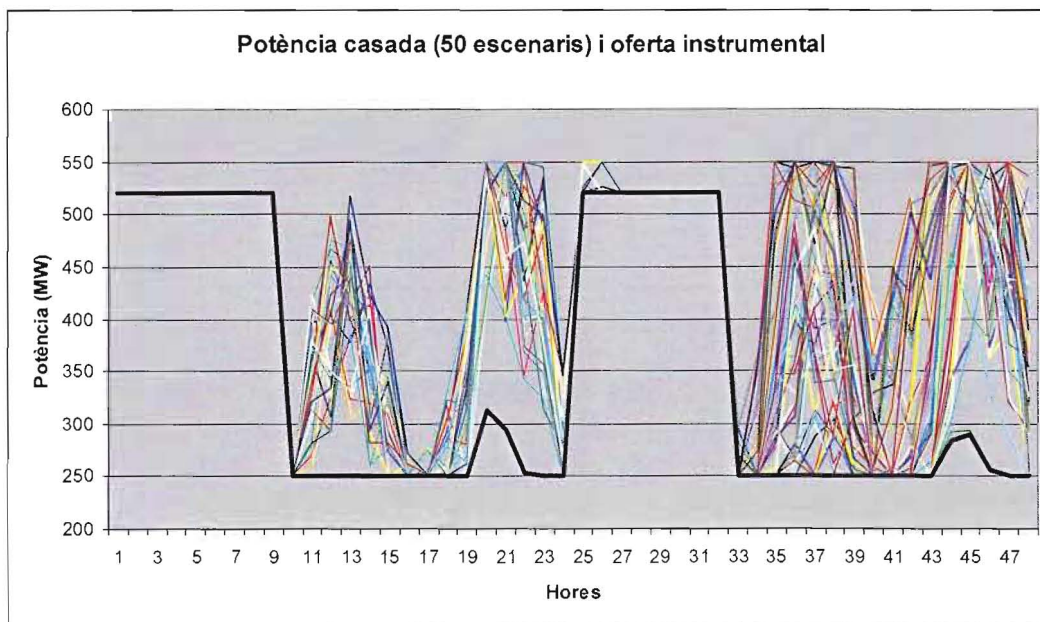


Figura 4.15: Potència cassada per 50 escenaris i oferta instrumental tèrmica 2

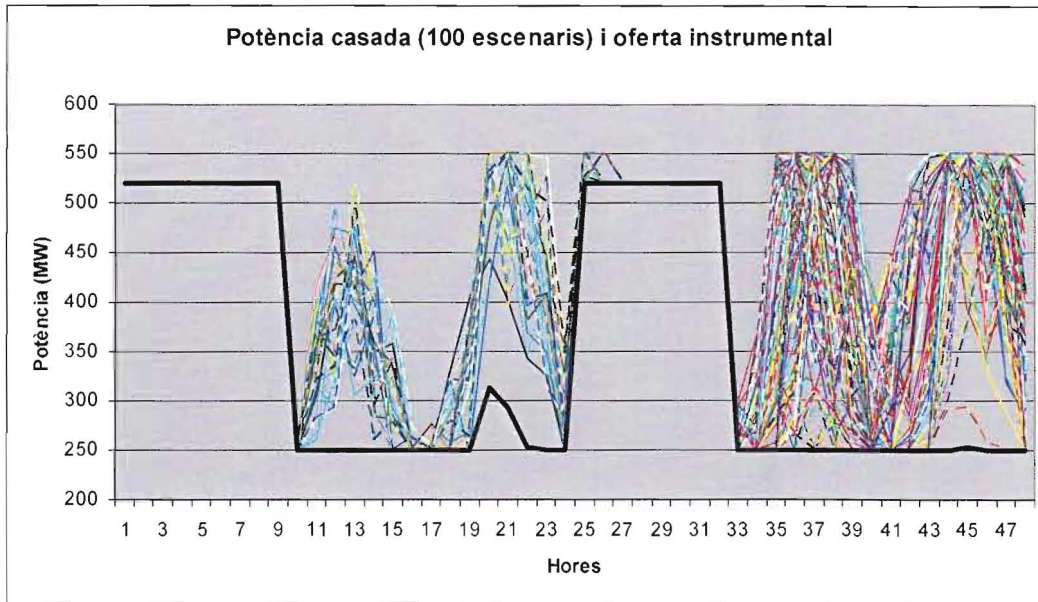


Figura 4.16: Potència cassada per 100 escenaris i oferta instrumental tèrmica 2

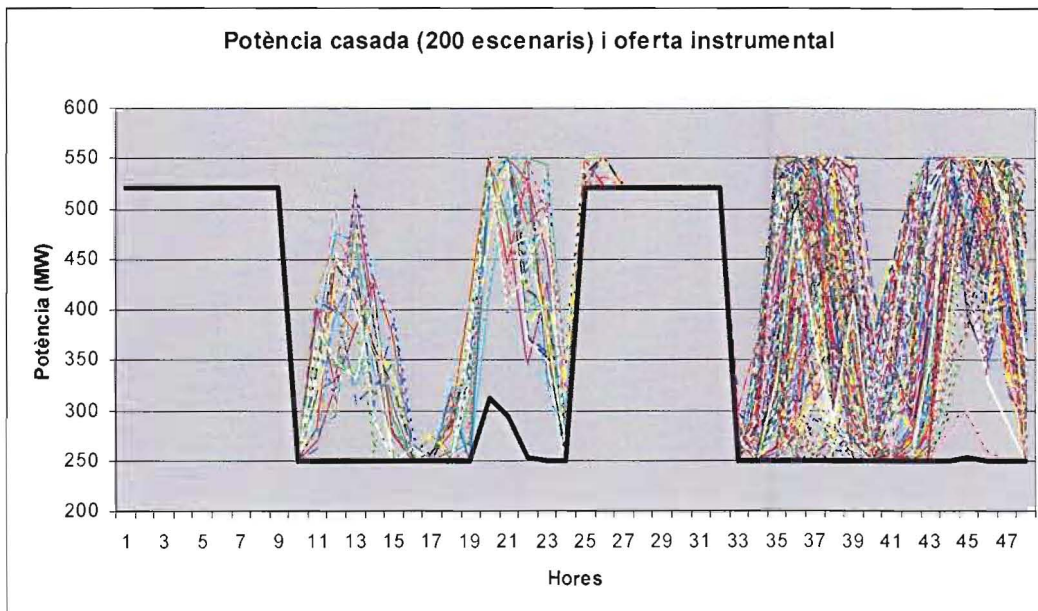


Figura 4.17: Potència cassada per 200 escenaris i oferta instrumental tèrmica 2

Es pot observar com el nombre d'escenaris, és a dir, la finesa de la representació de la variable aleatòria, afecta al resultat de la variable de decisió de primera etapa, la oferta instrumental (línea gruixuda). Els arbres de 10 i 50 escenaris proporcionen uns resultats clarament diferents als arbres de 100 i 200 escenaris, que proporcionen resultats idèntics. En aquest cas el nombre d'escenaris necessari, que seria el resultat de l'anàlisi d'estabilitat, estarà comprès entre 50 i 100.

5- Conclusions

5.1 Resum

El Projecte Final de Carrera s'emmarca dins el camp de la utilització de la Programació Estocàstica per a l'optimització de la generació d'energia. En particular, s'ha centrat en l'obtenció d'arbres d'escenaris que representen la possible evolució del preu de l'electricitat.

Normalment les tècniques utilitzades per a la construcció d'arbres generen arbres d'escenaris que contenen un nombre d'escenaris massa elevat, fet que comporta una resolució poc eficient del problema d'optimització. Per millorar l'eficiència, duem a terme una aproximació de l'arbre original amb un arbre format per un nombre més reduït d'escenaris.

Per tal d'assolir la reducció d'aquests arbres d'escenaris es van aplicar els mètodes *Simultaneous Backward Reduction* i *Fast Forward Selection* plantejats en l'article de N. Gröwe-Kuska, H. Heitsch, W. Römisch [10].

Els dos algorismes heurístics de reducció proposats determinen un subconjunt del conjunt d'escenaris inicials i assignen noves probabilitats als escenaris conservats. La metodologia de *Simultaneous Backward Reduction* es basa en l'eliminació d'escenaris fins a aconseguir el nombre desitjat d'escenaris conservats. Mentre que l'algorisme *Fast Forward Selection* es fonamenta en la selecció d'escenaris fins a obtenir el nombre escollit d'escenaris preservats.

Tot i que les tècniques implementades no requereixen de la resolució de cap problema d'optimització, es va usar el llenguatge de modelització AMPL perquè les variables de sortida que obtenim en l'execució dels algorismes són utilitzades posteriorment en la resolució de problemes d'optimització de l'oferta al mercat elèctric, que també han estat modelitzats en AMPL. D'aquesta forma el codi desenvolupat en aquest projecte es podria integrar fàcilment en els codis AMPL d'optimització de mercats elèctrics existents.

La implementació es va dividir, per una banda, en el codi de generació d'un arbre d'escenaris i, per l'altra, en el codi dels algorismes. Per a dur a terme l'execució dels algorismes necessitem el fitxer de comandes amb el codi dels algorismes i el fitxer de dades, el qual l'usuari ha de manipular per a obtenir el problema desitjat.

Per tal de comprovar el correcte funcionament de la implementació i millorar la seva eficiència es van realitzar un seguit de tests computacionals amb arbres amb preus reals o simulats, i amb el model d'optimització de l'oferta al mercat elèctric diari.

5.2 Consideracions finals

En la implementació portada a terme s'ha intentat utilitzar les estructures disponibles en AMPL de forma senzilla. Cal dir que AMPL és un llenguatge potent en el disseny de l'algorisme, i s'ha procurat aconseguir el codi més eficient en cada situació.

Cal destacar que el disseny de l'arbre es fonamenta en l'estructura d'un arbre de predecessors, i la seva generació ha ocupat un temps important de la implementació. Pel que fa a la implementació dels algorismes, ha estat més senzilla gràcies a que ja es disposava del seu pseudocodi.

Respecte el comportament dels algorismes s'extreuen les següents conclusions:

- *Fast Forward Selection* millora el temps d'execució quan el nombre d'escenaris conservats és petit, ja que quan ha de seleccionar un nombre menor d'escenaris realitza menys iteracions. Mentre *Simultaneous Backward Reduction* funciona millor quan la reducció no és tan forta, a causa de que el nombre de iteracions és més baix.
- A partir dels resultats del casos 1 i 2 podríem acceptar que l'algorisme de *Simultaneous Backward Reduction* obté un millor funcionament, pel que fa a correcció i eficiència.
- Pel que fa a l'ajust dels moments de la variable preu es pot observar que existeixen subconjunts d'escenaris que ajusten millor els moments de la variable que altres. Tot i que no sempre obtenim solucions òptimes, *Simultaneous Backward Reduction* i *Fast Forward Selection* proporcionen aproximacions raonablement bones de l'arbre original.

Respecte l'ús dels arbres reduïts als models d'optimització, es pot apreciar com el nombre d'escenaris de l'arbre té un efecte significatiu en la resolució del problema d'optimització de l'oferta al mercat elèctric diari.

5.3 Propostes a desenvolupar

A partir dels resultats del PFC es plantegen un seguit de futurs objectius:

- Millorar l'eficiència de la implementació dels mètodes.
- Realitzar nous tests computacionals amb arbres amb més etapes per tal d'estudiar el comportament del preu durant més dies.
- Crear una versió apta a futures utilitzacions o modificacions per part de nous usuaris.
- Manual detallat d'usuari.

6- Bibliografia

- [1] A.J. Conejo, F.J. Prieto. *Mathematical Programming and Electricity Markets*. Sociedad de Estadística e Investigación Operativa. Top 2001; 9:1-47.
- [2] OMEL: <http://www.omel.es/>
- [3] OMIP: <http://www.omip.pt/>
- [4] Red Eléctrica Española: <http://www.ree.es/>
- [5] Comisión Nacional de Energía: <http://www.cne.es/>
- [6] C. Corchero *Problema del període òptim d'oferta d'una unitat tèrmica*. PFC LCTE Cristina Corchero Garcia ; Director: F. J. Heredia ; Co-directora: Ma Pilar Muñoz. FME, UPC 2004.
- [7] F.J. Nogales, J. Contreras, A.J. Conejo, R. Espínola. *Forecasting Next-Day Electricity Prices by Time Series Models*. IEEE Transactions on Power Systems 2002; 17: 342-48.
- [8] J.R.Birge, F. Louveaux. *Introduction to Stochastic Programming*. Ed. Springer 1997.
- [9] E.Gómez-Villalva. *Gestión energética óptima de un consumidor industrial industrial de vapor y electricidad en mercados liberalizados*. Tesis Doctoral Emilio Gómez-Villalva. Director: A. Ramos. Escuela Técnica Superior de Ingeniería (ICAI), Departamento de Organización Industrial, Universidad Pontificia Comillas Madrid 2004, (www.iit.upcomillas.es/~aramos/tesis/tesis_EGV.pdf).
- [10] N. Gröwe-Kuska, H. Heitsch, W. Römisch: Scenario reduction and scenario tree construction for power management problems, IEEE Bologna Power Tech Proceedings (A. Borghetti, C.A. Nucci, M. Paolone eds.), 2003 IEEE.
- [11] AMPL: www.ampl.com/
- [12] P. L. Luque *Manual básico de AMPL*, 2000.
www.informatica.us.es/~calvo/ampl2a.pdf

7- Annexos

Annex 1

En aquest annex es presenta el codi AMPL necessari per generar un arbre d'escenaris. En aquest cas, un arbre amb 2 etapes i 300 escenaris de preus reals. Les dades de l'arbre complet es poden trobar en el CD adjunt al PFC.

Fitxer Dades

```
param T := 2;
param N:= 1 30 2 10;
param NJ:= 250;

param E: 1      2      3      4      5      6      7
:=
1      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
2      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
3      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
4      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
5      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
6      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
7      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
8      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
9      4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
10     4.07694  3.41953  2.39826  2.14628  2.11046  2.09882  2.11329
.....
290    5.08709  4.54789  2.6011  2.53565  2.53698  2.50079  2.30662
291    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
292    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
293    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
294    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
295    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
296    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
297    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
298    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
299    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
300    5.36671  3.79823  2.88701  2.54019  2.52975  2.361    2.64273
```

Comandes de generació de l'arbre

```
reset;
param T integer;
param N{1..T} integer;
param NJ integer;
param dimp integer;
param nod {1..T} integer;
param S integer;
param A {1..T,1..S};
set CN {1..T} within {1..dimp} ordered;
param cjt {1..dimp} integer;
set ant {1..dimp} within {1..S} ordered;
param ngrup integer;
set grup{1..ngrup}within {1..S} circular;
param pi {1..dimp};
param prob {1..S};
param preu {1..dimp,1..24};
param E {1..S,1..24*T};
data arbre300.dat;
```

```
let dimp := 0;
let S := 1;
for {i in 1..T}
{
  let S := N[i]*S;
  let nod[i] := S;
  let dimp := dimp + S;
}

let CN[1] := {1..N[1]};
for {i in 2..T}
{
  let CN[i] := {last(CN[i-1]) + 1..last(CN[i-1]) + nod[i]};
}
for {i in 1..T}
{
  for{j in first(CN[i])..last(CN[i])}
  {let cjt[j] := i;}
}

param aux1 integer;
for{t in 1..T}
{
  let aux1 := 1;
  for{k in 1..nod[t]}
  for{i in 1..(S / nod[t])}
  {
    let A[t,aux1] := member(k,CN[t]);
    let aux1 := aux1 + 1;
  }
}

param aux2 integer;
let aux2 := 0;
for{i in 1..dimp}
{
  if aux2 = S then let aux2 := 0;
  let ant[i] := {aux2 + 1..aux2 + (S / nod[cjt[i]])};
  let aux2 := last(ant[i]);
}

let ngrup:=S/N[T];
if T = 1 then {let grup[1]:= {1..N[T]}}
else{ for{i in 1..ngrup}let grup[i]:=ant[member(i,CN[T-1])];
};

param aux3 integer;
param aux4 integer;
let aux3 := 0;
for{t in 1..T}
  for{j in 1..(nod[t] / N[t])}
  {
    let aux4 := 1;
    for{m in aux3 + 1..aux3 + N[t]}
    {
      let pi[m] := 1/N[t];
      let aux4 := aux4 + 1;
    }
    let aux3 := aux3 + N[t];
  }
for{i in 1..S}
{
  let prob[i] := 1;
  for{t in 1..T}
  {
    let prob[i] := pi[A[t,i]] * prob[i];
  }
}
}
```


Annex 2

En aquest annex es presenta el codi AMPL necessari per executar l'algorisme Simultaneous Backward Reduction. L'arxiu amb el codi es pot trobar en el CD adjunt al PFC.

Algorisme 1 Simultaneous Backward Reduction

```
param ct {1..S,1..S};
param c {1..S,1..S};
set J {1..1} ordered;
set L{1..1} ordered;
param z {1..S};
set z_argmin{1..1} ordered;
set minim{1..1} ordered;
param q {1..S-NJ};
param nBScen integer;
set BScen{1..nBscen}within {1..S} circular;
param Sc {1..S-NJ,1..24*T};

for{i in 1..S,j in 1..S}
let ct[i,j] := sum{k in 1..24*T}{abs(E[i,k] - E[j,k])};

for{l in 1..S}
{
let c[l,l] := min {j in 1..S diff{l}} ct[l,j];
}
for{l in 1..S}
{
let z[l] := prob[l] * c[l,l];
}
let z_argmin[l]:= {i in 1..S: z[i] = min {j in 1..S} z[j]};
let J[l] := {first(z_argmin[l])};
let L[l] := 1..S diff {first(z_argmin[l])};
display first(z_argmin[l]) > info.res;

for{n in 2..NJ}
{
for{l in L[l]}
{
for{k in {l} union J[l]}
{
let c[k,l] := min {j in L[l] diff {l}} ct[k,j];
}
}
for{l in L[l]}
{
let z[l] := sum {k in {l} union J[l]}{prob[k] * c[k,l]};
}
let z_argmin[l]:= {i in L[l]: z[i] = min {j in L[l]} z[j]};
let J[l] := J[l] union {first(z_argmin[l])};
let L[l] := L[l] diff {first(z_argmin[l])};
display first(z_argmin[l]) > info.res;
}

for {i in J[l]}
{
let minim[l]:= {j in L[l]: ct[i,j] = min {k in L[l]} ct[i,k]};
let prob[first(minim[l])] := prob[first(minim[l])] + prob[i];
let prob[i] := 0;
display first(minim[l]) > info.res;
}
```

```
for(i in 1..ngrup)
{
let grup[i]:=grup[i] diff J[1];
}
let nBScen:=0;
for(n in 1..ngrup){
if(card(grup[n])!=0) then{let nBScen:=nBScen+1};
}
param aux6 integer;
param aux7 integer;
let aux6:=1;
let aux7:=1;
for(n in 1..ngrup){
if(card(grup[n])!=0) then{
for(i in first(grup[n])..last(grup[n]))
{
let BScen[aux7]:={aux6};
let aux6:=aux6+1;
}
}
let aux7:=aux7+1;
}
}
for(i in 1..S-NJ){
let q[i]:=prob[member(i,L[1])];}
for(i in 1..S-NJ){
for(j in 1..24*T)
let Sc[i,j]:=E[member(i,L[1]),j];
}

display {i in 1..T,j in L[1]}A[i,j] > back.res;
display J > back.res;
display prob > back.res;
display {i in L[1],j in 1..24*T}E[i,j] > back.res;
display nBScen > arbre.res;
display BScen > arbre.res;
display q > arbre.res;
display Sc > arbre.res;
close back.res;
close info.res;
close arbre.res;
quit;
```

Annex 3

En aquest annex es presenta el codi AMPL necessari per executar l'algorisme *Fast Forward Selection*. L'arxiu amb el codi es pot trobar en el CD adjunt al PFC.

Algorisme 2 Fast forward selection

```
param ct {1..S,1..S};
param c {1..S,1..S};
set J {1..1} ordered;
set L{1..1} ordered;
param z {1..S};
set z_argmin{1..1} ordered;
set minim{1..1} ordered;
param q {1..S-NJ};
param nBScen integer;
set BScen{1..nBScen}within {1..S} ordered;
param Sc {1..S-NJ,1..24*T};

for{i in 1..S}
for{j in 1..S}
{
  let ct[i,j] := sum{k in 1..24*T}(abs(E[i,k] - E[j,k]));
  let c[i,j] := ct[i,j];
}

for{u in 1..S}
{
  let z[u] := sum {k in 1..S diff {u}}(prob[k] * c[k,u]);
}
let z_argmin[1]:= {i in 1..S: z[i] = min {j in 1..S} z[j]};
let J[1] := 1..S diff {first(z_argmin[1])};
display first(z_argmin[1]) > info.res;

for{n in 2..(S - NJ)}
{
  for{k in J[1]}
  {
    for{u in J[1]}
    {
      let c[k,u] := min( c[k,u], c[k,first(z_argmin[1])] );
    }
  }
  for{u in J[1]}
  {
    let z[u] := sum{k in J[1] diff {u}}(prob[k] * c[k,u]);
  }
  let z_argmin[1]:= {i in J[1]: z[i] = min {j in J[1]} z[j]};
  let J[1] := J[1] diff {first(z_argmin[1])};
  display first(z_argmin[1]) > info.res;
}
let L[1]:=1..S diff J[1];

for {i in J[1]}
{
  let minim[1]:= {j in L[1]: ct[i,j] = min {k in L[1]} ct[i,k]};
  let prob[first(minim[1])] := prob[first(minim[1])] + prob[i];
  let prob[i] := 0;
  display first(minim[1]) > info.res;
}
```

```
for{i in 1..ngrup}
{
let grup[i]:=grup[i] diff J[1];
}
let nBScen:=0;
for{n in 1..ngrup}{
if(card(grup[n])!=0) then{let nBScen:=nBScen+1};
}
param aux6 integer;
param aux7 integer;
let aux6:=1;
let aux7:=1;
for{n in 1..ngrup}{
if(card(grup[n])!=0) then{
let BScen[aux7]:={};
for{i in 1..card(grup[n])}
{
let BScen[aux7]:=BScen[aux7]union{aux6};
let aux6:=aux6+1;
}
}
let aux7:=aux7+1;
}
}for{i in 1..S-NJ}{
let q[i]:=prob[member(i,L[1])];}
for{i in 1..S-NJ}{
for{j in 1..24*T}
let Sc[i,j]:=E[member(i,L[1]),j];
}

display {i in 1..T,j in L[1]}A[i,j] > for.res;
display J > for.res;
display prob > for.res;
display {i in L[1],j in 1..24*T}E[i,j] > for.res;
display nBScen > arbre.res;
display BScen > arbre.res;
display q > arbre.res;
display Sc > arbre.res;
close for.res;
close info.res;
close arbre.res;
quit;
```

Annex 4

En aquest annex es presenta els fitxers .mod i .dat corresponents al model d'optimització de l'oferta al mercat elèctric. Els arxius amb el codi es pot trobar en el CD adjunt al PFC.

Fitxer .mod

```
#
#Nombre intervals
#
param nI >0 integer;
set I := 1..nI ordered;
set Interv := 1..nI ordered;

#
#Parametres contractes Futurs
#
param nF integer;
set F := 1..nF ordered;
param L{F} >= 0 integer;

#
# Parametres escenaris i preus
#
param nScen integer;
set Scen := 1..nScen ordered;
param Prob{ Scen };
param PreuD{Interv, Scen };

#
# Nombre unitats
#
param T >0 integer;
set Term:= 1..T;

#
# Matriu que indica l'estat de cada unitat en cada interval de I
#
set TonI within Term cross I;

#
# Paràmetres unitats
#
param Cb {Term} ;
param Cl {Term} ;
param Cq {Term} ;
param Pmin {Term} >=0;
param Pmax {Term} >=0;

#
# Non-anticipativity
#
param nBScen integer; # number of scenario bundles
set BScen {1..nBScen} within {1..nScen} circular;

#####
# VARIABLES
#####
#
# Second stage (matched energy):
#
var p {(t,i) in TonI , Scen } <= Pmax[t];
```

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

```
#
# First stage (instrumental price bid):
#
var q {(t,i) in TonI} >= Pmin[t];
var qF {(t,i) in TonI, F} >= 0;

#####
# OBJECTIVE FUNCTION
#####

#####
# First period #
#####

minimize cost:
sum {(t,i) in TonI} (
  sum {s in Scen} Prob[s]*((Cl[t]-PreuD[i,s]*1000)*p[t,i,s] +
  Cq[t]*p[t,i,s]^2));

#####
# CONSTRAINTS
#####

#
# Constraints defining the matched energy (first period):
#
subject to cons_coupling_p_q { (t,i) in TonI, s in Scen}:
  p[t,i,s] >= q[t,i];
#
# Constraints coupling qI and qIF (first period):
#
subject to cons_coupling_q_qF { (t,i) in TonI }:
  q[t,i] >= sum{j in F} qF[t,i,j];
#
# Forward futures constraints (first period):
#
subject to cons_futurs { i in I, j in F}:
  sum{ (t,i) in TonI} qF[t,i,j] = L[j];

#
# Non-anticipativity
#
subject to cons_nonant { j in 1..nBScen, s in BScen[j], (t,i) in TonI :
i<=24 }:
  p[t,i,s] = p[t,i,next(s)];
```

Fitxer .dat

```
param nI:=48;
param nF:=2;

param L:=
1      500.0
2      1100.0;

param T:=9;

set TonI (tr):=
:      1  2  3  4  5  6  7  8  9  :=
1      -  +  +  +  -  +  +  -  -
2      -  +  +  +  -  +  +  -  -
3      -  +  +  +  -  +  +  -  -
4      -  +  +  +  -  +  +  -  -
5      -  +  +  +  -  +  +  -  -
```

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

6	-	+	+	+	-	+	+	-	-
7	-	+	+	+	-	+	+	-	-
8	-	+	+	+	-	+	+	-	-
9	-	+	+	+	-	+	+	-	-
10	+	+	+	+	+	+	+	-	+
11	+	+	+	+	+	+	+	-	+
12	+	+	+	+	+	+	+	-	+
13	+	+	+	+	+	+	+	-	+
14	+	+	+	+	+	+	+	-	+
15	+	+	+	+	+	+	+	-	+
16	+	+	+	+	+	+	+	-	+
17	+	+	+	+	+	+	+	-	+
18	+	+	+	+	+	+	+	-	+
19	+	+	+	+	+	+	+	-	+
20	+	+	+	+	+	+	+	-	+
21	+	+	+	+	+	+	+	-	+
22	+	+	+	+	+	+	+	-	+
23	+	+	+	+	+	+	+	-	+
24	+	+	+	+	+	+	+	-	+
25	-	+	+	+	-	+	+	-	-
26	-	+	+	+	-	+	+	-	-
27	-	+	+	+	-	+	+	-	-
28	-	+	+	+	-	+	+	-	-
29	-	+	+	+	-	+	+	-	-
30	-	+	+	+	-	+	+	-	-
31	-	+	+	+	-	+	+	-	-
32	-	+	+	+	-	+	+	-	-
33	-	+	+	+	-	+	+	+	-
34	+	+	+	+	-	+	+	+	+
35	+	+	+	+	-	+	+	+	+
36	+	+	+	+	-	+	+	+	+
37	+	+	+	+	-	+	+	+	+
38	+	+	+	+	-	+	+	+	+
39	+	+	+	+	-	+	+	+	+
40	+	+	+	+	-	+	+	+	+
41	+	+	+	+	-	+	+	+	+
42	+	+	+	+	-	+	+	+	+
43	+	+	+	+	-	+	+	+	+
44	+	+	+	+	-	+	+	+	+
45	+	+	+	+	-	+	+	+	+
46	+	+	+	+	-	+	+	+	+
47	+	+	+	+	-	+	+	+	+
48	+	+	+	+	-	+	+	+	+

param Cb:=
1 188849
2 537948
3 271328
4 408776
5 37869.3
6 457597
7 247414
8 180347
9 55909.900
;

param Cl:=
1 5046.04
2 3413.7
3 3584.13
4 3606.09
5 7583.1
6 -1715.58
7 4613.16
8 5620.67
9 5692.900
;

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

```
param Cq:=
1 1.4881
2 5.82283
3 5.40334
4 4.48861
5 0.
6 34.279
7 2.4543
8 0.
9 0.
;
```

```
param Pmin:=
1 160.00
2 250.00
3 80.00
4 160.00
5 30.00
6 60.00
7 160.00
8 90.00
9 110.00;
```

```
param Pmax:=
1 243.00
2 550.00
3 260.00
4 340.00
5 70.00
6 140.00
7 340.00
8 340.00
9 157.00;
```

```
param nScen := 50;
```

```
param Prob [*] :=
1 0.0333333 11 0.0333333 21 0.0133333 31 0.0166667 41 0.01
2 0.0333333 12 0.0333333 22 0.01 32 0.0133333 42 0.03
3 0.0166667 13 0.00333333 23 0.01 33 0.0166667 43 0.01
4 0.0166667 14 0.01 24 0.0166667 34 0.0333333 44 0.0166667
5 0.0333333 15 0.0133333 25 0.00666667 35 0.0333333 45 0.00666667
6 0.03 16 0.00666667 26 0.01 36 0.0333333 46 0.0166667
7 0.0333333 17 0.0133333 27 0.00666667 37 0.0333333 47 0.0166667
8 0.0333333 18 0.02 28 0.01 38 0.02 48 0.0333333
9 0.0333333 19 0.0133333 29 0.0166667 39 0.0133333 49 0.0166667
10 0.0333333 20 0.02 30 0.0166667 40 0.0333333 50 0.0166667
;
```

```
param PreuD [*,*] (tr)
: 1 2 3 4 5 6 7
:=
1 4.07694 3.41953 2.39826 2.14628 2.11046 2.09882 2.11329
2 5.08145 4.44519 3.05132 2.68 2.62355 2.61054 2.51866
3 5.54329 4.19614 2.69593 2.31533 2.41261 2.58452 2.4199
4 5.54329 4.19614 2.69593 2.31533 2.41261 2.58452 2.4199
5 5.24605 4.56686 2.63132 2.58045 2.34602 2.51598 2.62396
6 4.61686 4.21041 2.76334 2.66847 2.39845 2.37565 2.38581
7 5.91984 4.67381 2.77432 2.68887 2.39558 2.46776 2.53891
8 5.13298 4.24176 2.89198 2.45762 2.56251 2.44494 2.26942
9 5.54115 4.24977 2.92581 2.31462 2.28052 2.42428 2.46597
10 4.89164 4.12263 3.04009 2.46472 2.27592 2.44182 2.50709
11 5.81085 4.38563 2.87265 2.30803 2.51851 2.31204 2.29979
12 4.79728 4.70933 2.67884 2.45746 2.48534 2.43368 2.26125
13 4.79167 4.17453 2.84596 2.67539 2.40591 2.53107 2.34418
14 4.79167 4.17453 2.84596 2.67539 2.40591 2.53107 2.34418
15 4.79167 4.17453 2.84596 2.67539 2.40591 2.53107 2.34418
16 4.79167 4.17453 2.84596 2.67539 2.40591 2.53107 2.34418
```


Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

17	6.01494	4.49568	2.66879	2.37619	2.64455	2.3607	2.62505
18	6.01494	4.49568	2.66879	2.37619	2.64455	2.3607	2.62505
19	5.4734	4.16826	2.74938	2.60032	2.60014	2.32771	2.40561
20	5.4734	4.16826	2.74938	2.60032	2.60014	2.32771	2.40561
21	5.99382	4.139	2.7382	2.64491	2.61142	2.29833	2.42605
22	5.99382	4.139	2.7382	2.64491	2.61142	2.29833	2.42605
23	5.99382	4.139	2.7382	2.64491	2.61142	2.29833	2.42605
24	5.77957	4.0253	3.02412	2.61935	2.53046	2.5059	2.38908
25	5.77957	4.0253	3.02412	2.61935	2.53046	2.5059	2.38908
26	5.77957	4.0253	3.02412	2.61935	2.53046	2.5059	2.38908
27	4.89256	4.34955	2.80593	2.40066	2.46032	2.4616	2.61663
28	4.89256	4.34955	2.80593	2.40066	2.46032	2.4616	2.61663
29	4.89256	4.34955	2.80593	2.40066	2.46032	2.4616	2.61663
30	4.79641	3.902	2.77428	2.36638	2.45631	2.32822	2.27559
31	4.79641	3.902	2.77428	2.36638	2.45631	2.32822	2.27559
32	5.69757	4.21066	2.84098	2.40945	2.46735	2.47803	2.3428
33	5.69757	4.21066	2.84098	2.40945	2.46735	2.47803	2.3428
34	4.86291	4.30831	2.91869	2.55689	2.45244	2.31342	2.26198
35	5.61795	3.96673	2.60377	2.48949	2.26568	2.57586	2.59449
36	5.09932	4.75942	3.0229	2.62022	2.57828	2.34133	2.65798
37	4.66706	3.82724	2.7911	2.39863	2.4061	2.36961	2.33593
38	5.5872	4.31303	2.97683	2.54285	2.37395	2.5633	2.53947
39	5.5872	4.31303	2.97683	2.54285	2.37395	2.5633	2.53947
40	5.40577	4.05715	2.60412	2.43311	2.39298	2.521	2.33892
41	4.82252	4.36819	2.59795	2.46645	2.60777	2.49378	2.35401
42	4.82252	4.36819	2.59795	2.46645	2.60777	2.49378	2.35401
43	5.31666	4.60507	2.95029	2.4573	2.26699	2.39567	2.57718
44	5.31666	4.60507	2.95029	2.4573	2.26699	2.39567	2.57718
45	5.31666	4.60507	2.95029	2.4573	2.26699	2.39567	2.57718
46	4.86481	4.06987	2.99366	2.6621	2.63704	2.61755	2.37976
47	4.86481	4.06987	2.99366	2.6621	2.63704	2.61755	2.37976
48	5.08709	4.54789	2.6011	2.53565	2.53698	2.50079	2.30662
49	5.36671	3.79823	2.88701	2.54019	2.52975	2.361	2.64273
50	5.36671	3.79823	2.88701	2.54019	2.52975	2.361	2.64273

:	8	9	10	11	12	13	14
:=							
1	2.32238	2.7749	3.3459	4.14606	4.45669	4.55127	4.25004
2	2.82459	3.09736	4.52936	4.90774	5.38601	5.87925	6.26531
3	2.5414	3.36708	4.11101	5.92976	6.71271	6.6469	5.51622
4	2.5414	3.36708	4.11101	5.92976	6.71271	6.6469	5.51622
5	2.78458	3.35289	4.61598	5.64477	6.89016	6.17202	6.5018
6	2.86013	3.02411	4.2376	5.3048	5.12188	6.71942	6.09031
7	2.53273	3.25636	4.09009	5.66977	5.56554	5.49396	5.63388
8	2.71885	3.04829	4.47395	6.01866	6.08006	5.86387	6.34509
9	2.9568	3.031	4.33706	5.03592	6.46801	6.37402	5.74975
10	2.52778	3.19189	3.91714	5.27926	6.03332	5.39537	5.95477
11	2.9507	3.62488	4.39001	5.35473	5.44094	6.86716	5.92252
12	2.71685	3.06683	3.73353	6.07217	5.40513	7.09042	6.08952
13	2.76103	3.15466	3.77276	5.23931	6.22114	6.21758	5.71079
14	2.76103	3.15466	3.77276	5.23931	6.22114	6.21758	5.71079
15	2.76103	3.15466	3.77276	5.23931	6.22114	6.21758	5.71079
16	2.76103	3.15466	3.77276	5.23931	6.22114	6.21758	5.71079
17	2.75069	3.63996	4.05494	5.75417	5.82185	6.46694	5.61646
18	2.75069	3.63996	4.05494	5.75417	5.82185	6.46694	5.61646
19	2.81104	3.32191	3.96416	4.84032	6.15345	5.23391	5.37963
20	2.81104	3.32191	3.96416	4.84032	6.15345	5.23391	5.37963
21	2.92069	3.20601	4.16072	5.51568	6.54756	6.04164	5.65134
22	2.92069	3.20601	4.16072	5.51568	6.54756	6.04164	5.65134
23	2.92069	3.20601	4.16072	5.51568	6.54756	6.04164	5.65134
24	2.65974	3.66379	4.51966	6.1961	6.63449	6.52615	5.29623
25	2.65974	3.66379	4.51966	6.1961	6.63449	6.52615	5.29623
26	2.65974	3.66379	4.51966	6.1961	6.63449	6.52615	5.29623
27	2.85741	3.5605	4.09379	5.51713	6.91657	6.05859	6.16782
28	2.85741	3.5605	4.09379	5.51713	6.91657	6.05859	6.16782
29	2.85741	3.5605	4.09379	5.51713	6.91657	6.05859	6.16782
30	2.59878	3.61302	4.50132	5.75617	5.24459	7.08184	6.24619
31	2.59878	3.61302	4.50132	5.75617	5.24459	7.08184	6.24619

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

32	2.91256	3.0495	3.8422	6.26276	5.59571	6.78048	6.01054
33	2.91256	3.0495	3.8422	6.26276	5.59571	6.78048	6.01054
34	2.5935	3.37185	4.40442	5.50685	6.14185	5.39062	5.34923
35	2.87261	3.54556	4.53049	4.99695	5.29537	6.87828	4.81889
36	2.51829	3.14054	4.44547	5.74265	6.11163	6.83029	5.94617
37	2.88354	3.08865	3.82023	5.60435	6.49899	6.31952	5.0947
38	2.6585	3.06072	4.60612	5.82829	5.54118	5.91524	4.83521
39	2.6585	3.06072	4.60612	5.82829	5.54118	5.91524	4.83521
40	2.57184	3.2083	3.95239	5.4229	6.25724	6.41971	5.02895
41	2.84937	3.17956	4.10515	5.3727	5.48084	6.87994	5.13125
42	2.84937	3.17956	4.10515	5.3727	5.48084	6.87994	5.13125
43	2.72578	3.32141	3.94217	5.02797	5.12673	6.36715	6.03271
44	2.72578	3.32141	3.94217	5.02797	5.12673	6.36715	6.03271
45	2.72578	3.32141	3.94217	5.02797	5.12673	6.36715	6.03271
46	2.69245	3.23327	4.38588	6.14012	6.01152	6.41022	5.22324
47	2.69245	3.23327	4.38588	6.14012	6.01152	6.41022	5.22324
48	2.59379	3.50968	3.84209	5.5386	6.42896	6.19939	5.7253
49	2.83866	3.38375	4.07219	5.87795	5.62424	5.45897	6.14299
50	2.83866	3.38375	4.07219	5.87795	5.62424	5.45897	6.14299

:	15	16	17	18	19	20	21
:=							
1	4.01175	3.44751	3.4812	3.7269	4.09882	5.29943	5.13523
2	5.00074	4.26134	4.33484	5.33606	4.63823	8.55148	6.82827
3	5.10763	4.77374	4.97553	4.81946	4.89957	8.42629	8.11429
4	5.10763	4.77374	4.97553	4.81946	4.89957	8.42629	8.11429
5	4.60853	4.13378	4.00873	4.5443	5.82375	7.55742	8.3268
6	5.13038	4.19709	3.90175	4.38243	5.71434	8.45614	7.0729
7	5.09545	4.12392	3.87119	4.1919	5.29293	7.4522	6.79962
8	5.81904	4.13481	4.38734	5.11408	6.10667	7.77746	7.90836
9	4.58296	4.45898	4.77268	4.9862	5.74341	7.97867	8.45246
10	5.06054	4.15306	4.04683	5.39582	4.67411	6.22144	7.07675
11	5.20226	3.89362	4.22129	5.2142	4.64886	7.11416	7.66049
12	4.97729	4.48536	3.93207	4.82577	5.45174	7.85401	6.44604
13	5.49753	4.48687	4.97766	4.70706	5.49423	6.7207	7.30862
14	5.49753	4.48687	4.97766	4.70706	5.49423	6.7207	7.30862
15	5.49753	4.48687	4.97766	4.70706	5.49423	6.7207	7.30862
16	5.49753	4.48687	4.97766	4.70706	5.49423	6.7207	7.30862
17	5.68972	3.97641	4.03916	5.4485	6.10327	6.43059	6.07603
18	5.68972	3.97641	4.03916	5.4485	6.10327	6.43059	6.07603
19	5.57233	4.86648	4.31481	5.23751	5.37702	8.70541	7.37885
20	5.57233	4.86648	4.31481	5.23751	5.37702	8.70541	7.37885
21	4.69871	4.11847	4.28419	4.6642	5.4617	7.93904	8.00502
22	4.69871	4.11847	4.28419	4.6642	5.4617	7.93904	8.00502
23	4.69871	4.11847	4.28419	4.6642	5.4617	7.93904	8.00502
24	5.56575	4.58514	4.95972	4.74099	5.65799	8.58952	8.03922
25	5.56575	4.58514	4.95972	4.74099	5.65799	8.58952	8.03922
26	5.56575	4.58514	4.95972	4.74099	5.65799	8.58952	8.03922
27	4.68093	4.37063	4.49455	5.06477	5.01779	7.66043	6.2168
28	4.68093	4.37063	4.49455	5.06477	5.01779	7.66043	6.2168
29	4.68093	4.37063	4.49455	5.06477	5.01779	7.66043	6.2168
30	5.98899	4.93269	4.76239	4.34535	5.45744	7.09283	8.30747
31	5.98899	4.93269	4.76239	4.34535	5.45744	7.09283	8.30747
32	5.23811	4.63114	4.76448	4.20139	4.85831	7.73835	7.54133
33	5.23811	4.63114	4.76448	4.20139	4.85831	7.73835	7.54133
34	5.26942	4.28156	4.12634	5.20041	5.86763	8.76362	6.62187
35	5.39831	4.59046	3.94703	4.74347	5.1401	6.22093	8.52066
36	4.95546	4.49903	4.20608	4.66411	4.84831	7.56363	6.0303
37	4.6023	4.85154	4.41073	4.51791	5.876	7.28217	6.07016
38	4.99041	4.44474	4.14475	5.00004	4.86527	7.21094	7.54894
39	4.99041	4.44474	4.14475	5.00004	4.86527	7.21094	7.54894
40	5.02287	4.8493	4.76417	4.44673	5.65234	8.11189	7.72458
41	5.5291	4.01057	4.01644	4.44033	5.61756	8.00591	7.97262
42	5.5291	4.01057	4.01644	4.44033	5.61756	8.00591	7.97262
43	4.57161	3.95275	4.11254	5.38134	5.33697	7.12922	8.17407
44	4.57161	3.95275	4.11254	5.38134	5.33697	7.12922	8.17407
45	4.57161	3.95275	4.11254	5.38134	5.33697	7.12922	8.17407
46	5.93783	4.01505	4.42124	5.06894	5.56783	8.71123	8.40171

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

9	3.23137	2.5699	2.72971	3.40752	3.75695	4.04284	5.53573
10	3.05374	2.99647	2.6877	2.86745	4.05031	3.94242	5.17575
11	3.29627	3.04195	3.05461	2.9115	3.77691	3.44233	5.59575
12	3.11714	2.82212	2.68006	3.31631	3.94701	3.39919	5.08765
13	2.44722	2.40258	2.41163	2.82827	3.39678	3.68756	4.6778
14	2.85417	2.80259	2.83724	3.17224	4.83845	5.25048	6.02037
15	3.06933	3.08361	3.11398	3.65522	5.15155	4.95069	6.13482
16	2.99559	3.1496	2.70633	3.57187	3.8922	4.97873	6.63119
17	2.76809	3.24289	2.67657	3.53049	5.12357	4.52134	5.63799
18	3.28099	2.95326	2.7626	3.26839	4.27091	4.52291	7.24149
19	3.05881	2.64216	2.91523	3.54339	4.27586	4.57777	6.0705
20	2.89002	2.76229	3.08851	3.42103	4.01332	4.48208	7.97913
21	3.08497	2.91976	2.64733	3.61345	4.1474	5.05056	6.65053
22	3.1291	3.02015	3.07006	3.67296	4.55032	5.56371	7.17698
23	3.24978	2.6209	3.22472	3.60293	4.09735	5.3378	7.13708
24	3.19595	2.70143	3.25919	3.3423	5.15683	5.757	6.12745
25	2.83831	3.04191	3.13628	3.2803	4.56113	5.59235	7.73675
26	3.14314	2.91783	3.14811	3.8424	5.02525	4.4292	5.93222
27	3.33739	3.02208	3.11698	3.31614	4.58515	4.95541	7.96673
28	2.70188	3.24659	2.98927	3.50506	4.90688	4.25806	6.76341
29	2.69531	2.75566	3.15236	3.66638	4.50902	4.61767	5.95426
30	3.01401	2.85997	2.91065	3.59312	3.98526	5.60314	7.87357
31	3.08816	2.99892	3.19986	3.91655	3.86897	5.67125	6.94166
32	3.36601	2.77533	2.74211	3.15901	3.40986	2.86949	2.6345
33	3.0898	2.72026	2.80983	3.15112	3.74269	3.00758	2.71756
34	3.62905	2.96011	2.98475	3.44519	3.26671	3.28629	2.37377
35	3.30312	3.08708	2.83645	3.50747	3.02813	2.84441	2.44566
36	3.16716	3.00926	3.11357	3.28124	3.20125	2.7242	2.66891
37	3.16518	3.15469	2.66703	3.60597	3.40726	2.94008	2.37504
38	3.30102	3.20068	2.84076	3.32463	3.37845	2.90535	2.52827
39	3.54109	2.96688	2.87705	3.20262	3.87888	2.64823	2.26317
40	3.45939	3.06371	2.56599	2.93081	3.32116	2.66736	2.68074
41	2.38167	2.12062	2.12729	2.31072	2.38081	2.52369	3.50119
42	3.08218	2.86291	2.5445	2.94847	2.73558	3.60207	5.06588
43	2.97548	2.49566	2.55689	2.85948	3.22226	3.30787	4.37271
44	3.18112	2.73682	2.77516	2.69828	3.09506	3.55462	5.59128
45	2.69946	2.33331	2.65327	3.11903	2.71336	2.98528	4.27268
46	3.28521	2.88503	2.74834	2.65714	2.68872	3.05611	5.6008
47	3.03205	2.90932	2.73277	2.80231	3.39415	3.44927	5.15257
48	3.25179	2.654	2.66068	2.7876	2.98413	3.52108	5.19009
49	2.78557	2.88729	2.73649	3.31127	3.27626	3.47683	5.44315
50	2.96453	2.94277	2.78843	2.68029	3.00856	3.07646	4.34966

:	36	37	38	39	40	41	42
1	6.51618	5.96946	6.0421	5.17275	4.49821	4.18563	4.93128
2	5.3202	6.14383	5.31978	5.41755	4.94407	4.19675	5.35828
3	6.18073	5.23446	5.92831	4.63993	4.46578	4.69252	4.82811
4	5.73624	7.11778	6.2692	5.50964	3.97693	3.99024	4.32443
5	6.72515	5.75279	4.85429	5.16219	4.93425	4.68298	4.81062
6	5.68439	6.45803	6.42076	4.98765	4.82487	3.80774	4.04072
7	4.93645	6.36107	5.9056	4.79314	4.41273	4.10088	4.60167
8	6.03083	5.81699	6.45075	5.54109	4.17226	3.87427	4.90537
9	6.17296	5.71141	5.78888	5.15562	4.48058	4.8086	4.35182
10	4.66948	6.32561	5.25165	6.27362	4.39673	4.57544	4.86884
11	5.42408	5.76675	5.77421	6.10675	4.19567	4.18839	4.56946
12	5.37085	6.29052	6.86178	4.94089	5.05059	3.77143	4.99575
13	5.00814	5.26142	4.97407	4.41177	3.86713	3.99023	4.27032
14	6.18641	6.47886	7.02076	7.12181	6.05903	5.97095	6.13406
15	8.31081	8.98144	5.95463	6.00166	4.66815	5.30009	5.60213
16	8.58922	6.35371	7.75836	7.26804	5.78374	5.64135	5.20563
17	8.10171	7.1502	8.71964	6.98453	5.26774	4.65659	6.40874
18	7.32985	8.77987	7.30257	6.18308	5.65162	6.28176	5.88381
19	7.19421	7.92062	6.57782	5.60288	5.59823	5.11685	5.21752
20	8.62103	7.33291	6.42683	5.37171	5.00206	4.81992	5.40618
21	7.62502	6.98973	6.34335	6.6237	6.2019	5.66405	5.987
22	7.21025	9.45247	7.16896	6.86971	5.75468	6.39657	7.07154
23	8.18219	8.65718	7.33889	6.64657	5.19529	5.61593	7.01752

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

24	7.41164	9.29099	6.6376	5.70312	5.7541	5.51656	5.05387
25	7.31067	8.12713	7.96992	5.19879	4.51249	6.45695	6.27856
26	8.15449	6.53026	6.12248	6.68121	4.80777	6.15694	5.82129
27	8.29274	6.68375	7.62882	5.21497	5.44728	5.5058	6.32061
28	7.87619	8.15713	7.32546	7.30296	4.59166	6.48698	5.92488
29	7.91231	7.16295	8.65878	6.29427	4.79966	5.28471	6.5684
30	7.04045	6.98167	7.90567	6.68214	5.30884	4.73385	6.34213
31	7.55807	8.84688	8.5737	6.25881	5.53973	6.22331	6.94765
32	4.60326	5.07484	5.21753	4.35465	4.15644	3.86796	4.0143
33	4.84858	5.1756	4.79173	4.33913	3.40818	3.3019	3.62115
34	4.1496	4.41074	5.37219	4.5852	3.90532	3.55384	4.02751
35	3.95625	4.37382	4.45013	4.48748	4.0533	3.54566	3.36709
36	5.0101	4.49182	5.07026	4.00014	4.04917	3.1362	3.39286
37	4.59267	4.36237	5.17011	4.84816	3.37489	3.43634	4.12199
38	4.60083	4.58098	4.10134	4.45735	3.60803	3.883	3.82746
39	4.36125	5.24856	4.31921	3.87836	3.86486	3.6399	3.86428
40	4.87843	4.28768	4.72171	5.24197	3.80971	3.6942	4.13839
41	4.05554	4.24001	3.90145	3.80532	2.98734	2.90393	2.94444
42	5.23495	6.95769	6.60333	5.14643	4.3626	4.34307	3.75808
43	4.95527	7.08853	5.54739	6.10541	3.5537	3.69098	3.64195
44	5.59082	6.13921	6.33722	4.90263	4.56721	4.31434	3.55652
45	6.66079	5.51543	5.54096	4.89171	3.71803	3.68956	3.79907
46	5.96319	6.85671	5.87368	6.39008	3.96007	4.35483	3.69573
47	4.88234	5.29236	5.06535	4.80476	4.06687	4.40618	3.4966
48	6.88133	7.47089	6.01091	6.23595	4.32846	3.70372	3.52738
49	6.85623	5.79816	5.07661	6.14667	3.85449	3.65004	3.39142
50	6.46709	6.80479	5.61493	5.66712	4.68374	3.99252	4.01676

:	43	44	45	46	47	48	:=
1	5.7301	8.5098	7.91147	5.81668	6.27976	5.41715	
2	5.70397	7.00577	8.04166	7.17416	6.61748	5.33525	
3	5.90792	6.89787	7.54204	5.71774	5.99274	4.54465	
4	4.87754	8.76902	7.94314	7.18302	6.51409	4.72794	
5	5.5094	7.253	7.01515	6.7188	5.37041	5.14364	
6	5.07554	7.05967	7.44606	6.8873	6.83829	5.97471	
7	5.30011	6.45683	7.38826	7.03418	7.42537	5.99814	
8	5.1055	7.61835	7.32917	6.36538	6.42377	5.34386	
9	4.88074	8.06691	8.43288	7.44902	6.25686	5.20027	
10	5.2902	6.45352	6.67738	7.32287	6.33833	6.17872	
11	5.61279	7.17067	7.07939	7.19047	5.40333	5.08776	
12	4.58756	7.40927	8.67414	7.03671	6.1534	5.71276	
13	4.86432	6.53887	6.27958	5.4447	5.0697	4.31843	
14	7.05078	11.6509	12.4562	10.0839	7.70978	6.95247	
15	7.89616	10.3433	10.1623	8.86236	6.68764	5.75457	
16	7.33467	8.82712	9.9372	6.92963	8.75805	6.81567	
17	8.08403	8.38556	9.79601	8.9186	6.12347	5.6875	
18	7.58276	11.7248	10.6943	8.42335	6.92965	6.67251	
19	7.0369	10.3536	8.0514	7.65996	6.91382	5.82841	
20	8.24601	11.6447	10.7435	7.48472	8.78969	7.11485	
21	8.49914	11.6125	12.486	8.32299	8.33092	6.661	
22	5.98028	11.8854	7.91443	7.50998	7.91816	5.45729	
23	7.10075	11.4451	10.2857	10.105	6.54437	7.14541	
24	5.86683	12.4312	11.3389	8.26591	8.72815	5.69055	
25	6.84715	8.27944	8.52589	7.2182	8.55696	5.95344	
26	8.47503	11.9034	10.3017	6.92951	6.12298	6.32869	
27	7.32855	12.7245	11.8383	7.48589	9.08139	5.35756	
28	6.66608	13.1112	12.5708	8.49667	7.71274	6.53729	
29	7.21246	10.474	9.19806	10.0079	6.50583	5.76703	
30	8.49714	11.0889	9.85047	9.80879	7.68954	7.24096	
31	6.39219	12.6784	10.0105	6.90989	6.58759	5.43964	
32	4.26284	7.31167	8.02989	7.89462	6.79449	5.39714	
33	4.3774	5.27071	6.97613	5.73006	6.38375	6.35923	
34	3.76714	5.61916	7.59084	7.41346	6.9482	5.2445	
35	3.81959	6.61335	6.09765	5.87838	7.0826	5.17932	
36	3.94932	5.54291	5.87661	6.71004	7.30789	5.13738	
37	4.64369	5.43838	7.00743	7.64912	6.92185	5.88776	
38	4.10808	5.874	8.77019	6.26049	8.16529	4.66347	
39	4.14823	5.21334	5.8539	6.73453	7.15786	6.13308	

Reducció d'Escenaris per a l'Optimització de l'Oferta del Mercat Elèctric

40	3.92365	5.95786	7.51818	8.03176	6.27445	6.19364
41	3.37458	5.11182	5.1295	4.81576	4.30143	3.61829
42	4.70885	7.88888	7.95469	6.69475	6.10005	5.6608
43	4.43461	7.12686	9.12666	6.71464	7.17184	5.11659
44	4.60432	9.57113	8.6667	7.62951	6.95247	6.07298
45	5.20127	8.50774	6.84178	8.53105	5.8514	5.73058
46	5.55712	7.17506	9.44969	8.86321	5.67878	4.45515
47	5.59144	8.85936	8.32253	6.04252	7.25997	4.78235
48	5.44359	7.93312	9.07593	7.05235	5.61969	5.44688
49	5.61222	8.29012	9.79901	8.31008	6.94455	5.67974
50	4.54054	7.9391	8.15593	6.87942	6.34152	5.40153

;

param nBScen = 30;

```
set BScen[1] := 1;
set BScen[2] := 2;
set BScen[3] := 3 4;
set BScen[4] := 5;
set BScen[5] := 6;
set BScen[6] := 7;
set BScen[7] := 8;
set BScen[8] := 9;
set BScen[9] := 10;
set BScen[10] := 11;
set BScen[11] := 12;
set BScen[12] := 13 14 15 16;
set BScen[13] := 17 18;
set BScen[14] := 19 20;
set BScen[15] := 21 22 23;
set BScen[16] := 24 25 26;
set BScen[17] := 27 28 29;
set BScen[18] := 30 31;
set BScen[19] := 32 33;
set BScen[20] := 34;
set BScen[21] := 35;
set BScen[22] := 36;
set BScen[23] := 37;
set BScen[24] := 38 39;
set BScen[25] := 40;
set BScen[26] := 41 42;
set BScen[27] := 43 44 45;
set BScen[28] := 46 47;
set BScen[29] := 48;
set BScen[30] := 49 50;
```