Lenguajes algebraicos de modelado

Andrés Ramos

Alternativas desarrollo modelos optimización

- Lenguajes de programación de propósito general (C, C++, Java, VB, FORTRAN 90)
- Lenguajes o entornos de cálculo numérico o simbólico (hojas de cálculo, Matlab, Mathematica)
- Lenguajes algebraicos de modelado (GAMS, AMPL, AIMMS, XPRESS-MP)

Optimizadores en hojas de cálculo

- Ventajas
  - Fáciles de usar
  - Integración total con la hoja de cálculo
  - Familiaridad con el entorno que facilita la explicación del modelo y de sus resultados
  - Facilidad de presentación de resultados en gráficos

- Inconvenientes
  - No inducen una buena práctica de programación
  - Presentan dificultades para verificación, validación, actualización y documentación de los modelos
  - No permiten modelar problemas complejos o de gran tamaño

Biblioteca de optimización en C, C++

- Ventajas
  - Tiempo de solución es crítico
  - Permiten el uso de algoritmos de optimización específicos
  - Posibilidad de implantación del modelo en un entorno software o hardware especial
- Inconvenientes
  - Mayor dificultad y consumo de recursos para el mantenimiento del modelo

Ventajas lenguajes algebraicos (i)

- Lenguajes de alto nivel para formulación compacta de modelos grandes y complejos
- Facilitan desarrollo de prototipos
- Mejorar productividad de modeladores
- Estructuran buenos hábitos de modelado
- Separan datos de estructura matemática de modelo
- Formulación independiente del tamaño
- Modelo independiente de optimizadores

Ventajas lenguajes algebraicos (ii)

- Facilitan reformulación continua
- Documentación simultánea al modelo
- Permiten implantación de algoritmos avanzados
- Implantación fácil de problemas NLP, MIP, MCP
- Portabilidad entre plataformas y sistemas operativos (Windows, Linux, Sun Solaris, HP UX, DEC Digital Unix, IBM AIX, SGI IRIX)

Desventajas lenguajes algebraicos

- No son adecuados para usos esporádicos con problemas de pequeño tamaño
- No son adecuados para resolución directa problemas de tamaño gigantesco (1.000.000 x 1.000.000)

Tendencias futuras

- Interfaz visual en formulación
- Interfaz más estrecha con hojas de cálculo y bases de datos
- Interfaz con funciones externas escritas en lenguajes de propósito general
- Resolución directa de problemas optimización estocástica (OSLSE, DECIS)
- Selección automática de método de optimización y del optimizador

Aplicaciones reales

- En IIT se pasó de utilizar FORTRAN a utilizar GAMS exclusivamente
- Problemas de hasta 117.000 restricciones, 225.000 variables y 655.000 elementos no nulos resueltos con facilidad
- Incorporación de algoritmos avanzados (descomposición anidada estocástica de Benders) en modelos

Modelo de transporte en GAMS (i)

Sean *i* fábricas de envasado y *j* mercados de consumo. Cada planta de producción tiene una capacidad máxima de *ai* cajas y cada mercado demanda una cantidad *bj* de cajas (se supone que la capacidad de producción total de las fábricas es superior a la demanda total para que el problema sea factible). El coste de transporte entre cada fábrica *i* y cada mercado *j* por cada caja es *cij*. Se desea satisfacer la demanda de cada mercado al mínimo coste. Las variables de decisión del problema serán las cajas transportadas entre cada fábrica *i* y cada mercado *j*, *xij*.

Formulación matemática

- Función objetivo

$$\min \sum_i \sum_j c_{ij} x_{ij}$$

- Límite de producción de cada fábrica i

$$\sum_j x_{ij} \le a_i \quad \forall i$$

- Consumo de cada mercado j

$$\sum_i x_{ij} \ge b_j \quad \forall j$$

- Cantidad a enviar desde cada fábrica i a cada mercado j

$$x_{ij} \ge 0 \quad \forall i \to j$$

Modelo de transporte en GAMS (i)

```
SETS
   I fábricas de envasado / VIGO, ALGECIRAS /
   J mercados de consumo  / MADRID, BARCELONA, VALENCIA / ;

PARAMETERS
   A(i) capacidad de producción de la fábrica i [cajas]
       / VIGO      350
         ALGECIRAS 700 /

   B(j) demanda del mercado j [cajas]
       / MADRID    400
         BARCELONA 450
         VALENCIA  150 / ;

TABLE C(i,j) coste transporte entre i y j [miles de pesetas por caja]
          MADRID BARCELONA VALENCIA
VIGO        0.06      0.12      0.09
ALGECIRAS   0.05      0.15      0.11 ;
```

Modelo de transporte en GAMS (ii)

```
VARIABLES
   X(i,j) cajas transportadas entre fábrica i y mercado j [cajas]
   CT     coste de transporte [miles de pesetas] ;

POSITIVE VARIABLE X ;

EQUATIONS
   COSTE        coste total de transporte   [miles de pesetas]
   CAPACIDAD(i) capacidad máxima de cada fábrica i     [cajas]
   DEMANDA(j)   satisfacción demanda de cada mercado j [cajas] ;

COSTE ..        CT =E= SUM((i,j), C(i,j) * X(i,j)) ;

CAPACIDAD(i) .. SUM(j, X(i,j)) =L= A(i) ;

DEMANDA(j) ..   SUM(i, X(i,j)) =G= B(j) ;

MODEL TRANSPORTE / COSTE, CAPACIDAD, DEMANDA / ;

SOLVE TRANSPORTE USING LP MINIMIZING CT ;
```

Biblioteca de problemas

- Gestión de la producción
- Economía agraria
- Ingeniería química
- Ingeniería de montes
- Comercio internacional
- Desarrollo económico
- Micro y macroeconomía
- Modelos de equilibrio generalizado
- Economía de la energía
- Finanzas
- Estadística, econometría
- Investigación operativa

Programación

- Disciplina cuyo dominio es básico en muchos proyectos de ingeniería
  - Ciencia: pensamiento, disciplina, rigurosidad y experimentación
  - Arte: belleza y elegancia
- Aprendizaje mediante la lectura
- Un buen diseño es fundamental
- Programar por refinamiento gradual
- Utilizar una maqueta para el desarrollo, depuración y verificación de un modelo
- Cuidado con los detalles

Elementos de estilo de programación (calidad)

- Modularidad: datos, estructura modelo y resultados en diferentes ficheros
- Mismo uso en diferentes partes del código
- Comprobación del dominio de los índices

Elementos de estilo de programación (mantenibilidad, reusabilidad)

- Código limpio, cuidar la estética en la escritura
- Código debe ser autodocumentado
- Mantener la coherencia de las reglas de escritura (endentación en instrucciones repetitivas)
- Facilitar su lectura (paralelismo entre instrucciones consecutivas similares, indentación)
- Comentarios ilustrativos y bien localizados
- Nombres descriptivos de identificadores
- Definiciones con dimensiones físicas
- Referencias al origen de los datos
- Uso sistemático y coherente de mayúsculas y minúsculas

GAMS (General Algebraic Modeling System)

- Entorno de desarrollo GAMSIDE

- Manual de usuario Help-Docs-gamsusersguide.pdf
- Manuales de optimizadores Help-Docs-solvers

- Modelo: nombre_fichero.gms
- Resultados: nombre_fichero.lst
- Registro del proceso: nombre_fichero.log

Formato general de las instrucciones GAMS

- Líneas con * en primera columna son de comentario
- No se distingue entre mayúsculas y minúsculas
- El paréntesis (), el corchete [] o la llave {} se pueden utilizar para separar niveles entre sí con igual funcionamiento
- Palabras reservadas del lenguaje aparecen resaltadas
- Instrucciones acaban en un ; (que puede eliminarse cuando la siguiente palabra sea reservada)

Estructura general de un modelo

- Declaración de conjuntos y parámetros. Asignación de valores por omisión.
- Inclusión y manipulación de datos de entrada. Parámetros auxiliares
- Variables
- Ecuaciones
- Modelo
- Acotamiento e inicialización de variables
- Resolución del problema de optimización
- Presentación de resultados

Bloques de un modelo en GAMS

- Obligatorios
  `VARIABLES`
  `EQUATIONS`
  `MODEL`
  `SOLVE`

- Opcionales
  `SETS`: (`ALIAS`)
  `DATA`: `SCALARS`, `PARAMETERS`, `TABLE`

- Los valores de `INF`, `EPS` son válidos como datos

VARIABLES

- Siempre debe haber una variable libre para representar el valor de la función objetivo
- Tipos:

  FREE (por omisión) -∞ a + ∞

  POSITIVE 0 a + ∞

  NEGATIVE - ∞ a 0

  BINARY 0 ó 1

  INTEGER 0 a 100

- Sufijos:
  – nombre_var.LO cota inferior
  – nombre_var.UP cota superior
  – nombre_var.L valor inicial antes y valor óptimo después
  – nombre_var.M valor marginal (coste reducido)
  – nombre_var.FX fija una variable a un valor

EQUATIONS

- Bloques:
  - Declaración con comentario explicativo
  - Expresiones matemáticas

- Tipos: =E= $=$, =L= $\leq$, =G= $\geq$

- Sufijos:
  - nombre_ec.LO cota inferior
  - nombre_ec.UP cota superior
  - nombre_ec.L valor inicial antes y valor óptimo después
  - nombre_ec.M valor marginal (variable dual o precio sombra)

MODEL y SOLVE

- MODEL nombre_modelo / nombre_ecuaciones /

  MODEL nombre_modelo / ALL /

- SOLVE nombre_modelo USING tipo_problema MINIMIZING (MAXIMIZING) variable_f.o.

Tipos de problemas y optimizadores

- LP, RMIP (programación lineal): **BDMLP**
- MILP (programación lineal entera mixta): **CPLEX, OSL, XA, XPRESS, ZOOM**
- NLP (programación no lineal): **CONOPT, MINOS, SNOPT**
- DNLP (programación no lineal con derivadas no continuas): **CONOPT, MINOS, SNOPT**
- MINLP (programación no lineal entera mixta): **DICOPT, SBB**
- SP (programación estocástica): **DECIS, OSLSE**
- MCP (problema mixto complementario): **MILES, PATH**
- MPEC (programación matemática con restricciones de equilibrio):
- CNS (sistemas no lineales restringidos): **CONOPT, PATH**
- MPSGE (análisis de equilibrio generalizado)

Operador $ en asignaciones, sumatorios, restricciones

- Establece una condición

  $(VALOR > 0) $(I NE J)

- A la izquierda de una asignación: Realiza la asignación SÓLO cuando se cumple la condición

```
IF (condición,
    REALIZA LA ASIGNACIÓN
) ;
```

- A la derecha de una asignación: Realiza SIEMPRE la asignación tomando ésta el valor 0 si no se cumple la condición

```
IF (condición,
    REALIZA LA ASIGNACIÓN
ELSE
    ASIGNA VALOR 0
) ;
```

Operaciones relacionales

- `LT` <, `GT` >, `EQ` =, `NE` <>, `LE` <=, `GE` >=

- `NOT`, `AND`, `OR`, `XOR`

- `DIAG(elemento_conjunto,elemento_conjunto)`={1,0}
- `SAMEAS(elemento_conjunto,elemento_conjunto)`={V,F}

Operaciones con conjuntos

- Intersección

  `D(a) = B(a) * C(a)`
- Unión

  `D(a) = B(a) + C(a)`
- Complementario

  `D(a) = NOT C(a)`
- Diferencia

  `D(a) = B(a) - C(a)`

Funciones

- Elementales: +, -, *, /, ** ó POWER(x,n)
- ORD, CARD ordinal y cardinal de un conjunto
- Con índices: SUM, PROD, SMAX, SMIN
- Otras funciones: ABS, ARCTAN, SIN, COS, CEIL, FLOOR, EXP, LOG, LOG10, MAX, MIN, MOD, ROUND, SIGN, SQR, SQRT, TRUNC, NORMAL, UNIFORM
- Funciones de tiempo: GYEAR, GMONTH, GDAY, GHOUR, GMINUTE, GSECOND, GDOW, GLEAP, JDATE, JNOW, JSTART, JTIME

Conjuntos dinámicos

- Subconjuntos de conjuntos estáticos cuyo contenido puede cambiar mediante asignaciones

```
SETS M meses / 1 * 12 /
     MP(m) meses pares ;
display m ;
MP(m) $[MOD(ord(m),2) = 0] = YES ;
display mp ;
MP('3') = yes ;
display mp ;
MP(m) $(ord(m) = 3) = NO ;
display mp ;
```

- Elementos fundamentales en el desarrollo de modelos en GAMS
- Deben utilizarse sistemáticamente para evitar formular ecuaciones o variables o asignaciones innecesarias.

Modelo de transbordo

- Sea un conjunto de nodos conectados mediante arcos. Un nodo no tiene que estar conectado con todos los demás nodos. Un nodo puede ser de generación, de demanda o de transbordo según produzca, consuma o trasvase el producto. La oferta total es mayor o igual que la demanda total. Se supone conocida la capacidad máxima de oferta y la demanda de cada nodo. También se conoce el coste unitario de transporte de producto para cada arco. Se trata de minimizar el coste de transporte satisfaciendo la demanda.

- Extensión del problema anterior añadiendo arcos no dirigidos.

Desplazamientos de índices. Adelantos y retrasos

- ```
  t=E,F,M,A,MY,J,JL,AG,S,O,N,D
  RVA(t) + APOR(t) - GASTO(t) =E= RVA(t+1)
  ```

- Valores de vectores fuera del dominio son 0
  ```
  RVA('D') + APOR('D') - GASTO('D') =E= 0
  ```

- Recorrido circular de un índice
  ```
  t=1,...,12
  RVA(t) + APOR(t) - GASTO(t) = RVA(t++1)
  RVA('12') + APOR('12') - GASTO('12') = RVA('1')
  ```

- Recorrido en orden inverso del índice de `PP` aun cuando `i` se recorre en sentido creciente
  ```
  PP(i+[card(i)-2*ord(i)+1])
  ```
  ```

Contratación de vendedores

La sección de venta de billetes de una estación de metro necesita las siguientes personas durante las 24 horas del día

Intervalo	Vendedores
00 – 06	2
06 – 10	8
10 – 12	4
12 – 16	3
16 – 18	6
18 – 22	5
22 – 24	3

Cada vendedor trabaja ocho horas en dos bloques de cuatro con una hora de descanso al cabo del primer bloque. El turno puede empezar en cualquier hora del día. Determinar el número mínimo de vendedores a contratar.

Ejercicios de adelantos y retrasos

- Distancias entre cruces
  - Suponer una ciudad completamente reticulada con una longitud unitaria de cada lado de la retícula. Calcular analíticamente la distancia entre dos cruces cualesquiera de la ciudad.
- Máximo número de caballos
  - Determinar mediante un problema de optimización el máximo número de caballos que pueden estar en un tablero de ajedrez sin comerse entre sí
- Máximo número de reinas
  - Determinar mediante un problema de optimización el máximo número de reinas que pueden estar en un tablero de ajedrez sin comerse entre sí

TABLES (i)

- Continuación de tablas con múltiples columnas

```
SETS i / MAD, BCN /
     j / A1, A2, A3, A4, A5, A6 /

TABLE CAPACIDAD(i,j) capacidad máxima
           A1   A2   A3
MAD        1    0    3
BCN         2    1    2
+
           A4   A5   A6
MAD        2    1    3
BCN         3    2    2
```

TABLES (ii)

- Tablas con más de dos dimensiones

```
SETS i / MAD, BCN /
     j / A1, A2, A3, A4, A5, A6 /
     K / A, B, C /
TABLE CAPACIDAD(i,j,k) capacidad máxima
           A   B   C
MAD.A1     1   0   3
MAD.A2     2   1   2

TABLE CAPACIDAD(i,j,k) capacidad máxima
           A1.A  A1.B  A1.C  A2.A  A2.B
MAD          1     0     3     6     8
BCN          2     1     2     2     4
```

Repeticiones

- LOOP (conjunto,
) ;
- WHILE (condición,
) ;
- IF (condición,
  ELSE
) ;
- REPEAT
  UNTIL condición;
- FOR (i=inicio TO/DOWNTO final BY incremento,
) ;

Entrada/salida de datos

- Entrada de datos por fichero

  `$include` nombre_del_fichero

- `DISPLAY` nombre_identificador

- Salida de datos por fichero

  `file` nombre_interno / nombre_externo /

  `put` nombre_interno

  `put` nombre_identificador

  `putclose` nombre_interno

- Existen opciones específicas de control de formato de la salida

Estructura general de un modelo “comercial”

- Declaración de conjuntos y parámetros. Asignación de valores por omisión.
- Variables
- Ecuaciones
- Modelo

- Inclusión y manipulación de datos de entrada. Parámetros auxiliares
- Acotamiento e inicialización de variables
- Resolución del problema de optimización
- Presentación de resultados

Modelo de transporte en GAMS (i)

```
SETS
   I fábricas de envasado
   J mercados de consumo

PARAMETERS
   A(i)   capacidad de producción de la fábrica i [cajas]
   B(j)   demanda del mercado j [cajas]
   C(i,j) coste transporte entre i y j [miles de pesetas por caja]

VARIABLES
   X(i,j) cajas transportadas entre fábrica i y mercado j [cajas]
   CT     coste de transporte [miles de pesetas] ;

POSITIVE VARIABLE X ;
```

Modelo de transporte en GAMS (ii)

```
EQUATIONS
   COSTE          coste total de transporte   [miles de pesetas]
   CAPACIDAD(i) capacidad máxima de cada fábrica i     [cajas]
   DEMANDA(j)   satisfacción demanda de cada mercado j [cajas] ;

COSTE ..        CT =E= SUM((i,j), C(i,j) * X(i,j)) ;

CAPACIDAD(i) .. SUM(j, X(i,j)) =L= A(i) ;

DEMANDA(j) ..   SUM(i, X(i,j)) =G= B(j) ;

MODEL TRANSPORTE / COSTE, CAPACIDAD, DEMANDA / ;

$include datos.gms

SOLVE TRANSPORTE USING LP MINIMIZING CT ;
```

Modelo de transporte en GAMS (iii)

```
* introducción de datos de entrada
SETS
   I fábricas de envasado / VIGO, ALGECIRAS /
   J mercados de consumo  / MADRID, BARCELONA, VALENCIA / ;
PARAMETERS
   A(i) capacidad de producción de la fábrica i [cajas]
       / VIGO      350
         ALGECIRAS 700 /
   B(j) demanda del mercado j [cajas]
       / MADRID    400
         BARCELONA 450
         VALENCIA  150 / ;
TABLE C(i,j) coste transporte entre i y j [miles de pesetas por caja]
          MADRID BARCELONA VALENCIA
VIGO       0.06     0.12     0.09
ALGECIRAS  0.05     0.15     0.11 ;
```

Modularidad y ocultación de código

- Separar formulación de un problema de sus datos. Proteger la confidencialidad de la formulación

- Versión RUNTIME de un modelo

  SAVE y RESTART

- Secure Work Files

  Permiten controlar el acceso a símbolos en particular y crear ficheros de rearranque (restart) seguros asociados a licencias GAMS particulares

- Funciones dentro de un código

Cualificadores de ejecución

- `SUPPRESS 1`

  suprime el eco del listado del código
- `PW 94 PS 999999`

  especifica la anchura y longitud de la página
- `CHARSET 1`

  admite caracteres internacionales en las definiciones

Directivas $, opciones OPTIONS, cualificadores de modelo (i)

- `$`nombre_directiva
- `OPTION` nombre_opción
- nombre_modelo`.cualificador`

- `$ON(OFF)SYMLIST`, `ON(OFF)SYMXREF`
  lista y tabla de referencias entre los símbolos del código
- `OPTION LIMROW`=número_filas_vistas
- `OPTION LIMCOL`=número_columnas_vistas
- `OPTION SOLPRINT=ON(OFF)` permite o suprime la información sobre la solución óptima
- nombre_modelo.`SOLPRINT`= 0,1,2
- `OPTION DECIMALS`=número_decimales en DISPLAY

Directivas $, opciones OPTIONS, cualificadores de modelo (ii)

- nombre_modelo.`OPTFILE`= 1,2,...

  Parámetros en ficheros `cplex.opt`, `cplex.op2`, ...
- `OPTION SYSOUT=ON(OFF)` escribe fichero de opciones del optimizador
- `OPTION ITERLIM`=número_máx_iteraciones
- `OPTION RESLIM`=tiempo_ejecución_máx

Directivas $, opciones OPTIONS, cualificadores de modelo (iii)

- `OPTION SOLVEOPT=REPLACE` reemplaza los valores de la solución
- `$CLEAR`=nombre_parámetro
- nombre_modelo.`SOLSLACK` presenta las variables de holgura de las restricciones
- nombre_modelo.`HOLDFIXED` elimina del problema las variables con valores fijos

- `OPTION SEED`=número permite fijar la semilla del generador de números aleatorios

Interfaces

- GAMS Convert
  - Transforma un modelo GAMS en un formato utilizable por otros sistemas de modelado u optimizadores
- GDX (GAMS Data Exchange)
  - Permite el intercambio de datos con una hoja de cálculo
- Escritura en formato CSV

  `$libinclude gams2txt` nombre_identificador
- Matlab

Estilo de programación

Andrés Ramos

Uso avanzado de GAMS

- minimización del tiempo de ejecución y/o de la memoria
- importante cuando se trata de problemas de muy gran tamaño (> 100.000 x 100.000) o resolución iterativa de numerosos problemas (más de 100)
- aparece al usar simulación de Monte Carlo o técnicas de descomposición

Tiempo de ejecución de modelos escritos en GAMS

- tiempo de creación
  - formulación del problema específico
- tiempo de interfaz
  - comunicación entre lenguaje GAMS y optimizador
- tiempo de optimización
  - resolución del problema por el optimizador

Análisis de consumos de tiempo/memoria

- dependiente del tamaño y estructura de la matriz de restricciones
- número de resoluciones (iteraciones)
- variación entre soluciones sucesivas de los parámetros estocásticos

Control del tiempo y del espacio

- `OPTION PROFILE, PROFILETOL`
- Cualificador `STEPSUM`
- `ABORT` condición
- `OPTION BRATIO`=número
- Los valores de las variables son guardados siempre.
- La supresión de la información de salida en el nombre_fichero.lst se consigue con las siguientes opciones.

```
$OFFSYMLIST, OFFSYMXREF, OFFUELLIST, OFFUELXREF
OPTION LIMROW=0, LIMCOL=0, SOLPRINT=OFF, SYSOUT=OFF
nombre_modelo.SOLPRINT=2 ;
```

  y escribiendo en la invocación de GAMS

```
gams nombre_modelo.gms suppress 1
```

  Además, también se puede suprimir la información en pantalla que produce el optimizador con los consiguientes parámetros (por ejemplo, para CPLEX `simdisplay 0 bardisplay 0 mipdisplay 0`).

```
gams nombre_modelo.gms ll 0 lo 0
```

Direcciones de mejora

- informáticas (asociadas al lenguaje GAMS)
- matemáticas (reformulación del problema)

- afectan conjuntamente al tiempo de ejecución
- criterios dependientes del problema, indican direcciones a explorar

Direcciones a explorar (i)

- uso de disco virtual (RAMDRIVE)
- selección de optimizador y algoritmo de optimización y uso de últimas versiones
- cambio en orden de índices en instrucciones de asignación, formulación de ecuaciones
- modelado de conjunto ordenados SOSn

Direcciones a explorar (ii)

- ajuste de parámetros del optimizador
- uso de bases previas (parámetro BRATIO)
- mejoras en el barrido de las numerosas optimizaciones
- detección de infactibilidades (parámetro iis de CPLEX)
- análisis de sensibilidad (disponible en CPLEX y OSL)

Comparación entre optimizadores y método de optimización

	Tiempo	p.u.	Iter	Tiempo	p.u.	Iter
CPLEX 6.0						
Punto interior	41.8	1.0	32	237.3	1.0	35
Simplex dual	99.8	1.4	12692	1812.6	6.6	48695
Simplex primal	156.2	3.7	21622	1217.5	5.1	50280
MINOS 5.3						
Simplex primal	1863.6	44.6	23927	---	---	---
OSL 2.1						
Punto interior	163.9	3.9	10798	774.4	3.3	19524
Simplex primal	530.9	12.7	12685	7426.6	31.3	62019

Tiempo para un ordenador Pentium II con 128 MB a 233 MHz

Selección de optimizador y método de optimización en LP

- No existe una regla clara
- No hay regla para determinar qué algoritmo simplex es más eficiente. Muy sensible a la estructura del problema
- Probar y observar

Método simplex	Hasta 10.000 x 10.000
Método simplex	Análisis de sensibilidad, problemas MIP
Método de punto interior	Desde 10.000 x 10.000 hasta 100.000 x 100.000
Métodos de descomposición	Más de 100.000 x 100.000

Cambio en instrucciones de asignación

- orden de los índices consistente en todos los elementos del modelo

```
PP(i,j,k) = QQ(k,j,i) * 1.1 ;   NO
PP(i,j,k) = QQ(i,j,k) * 1.1 ;   SI
```

- orden de barrido en instrucciones reiterativas coherente
- hacer uso extensivo de condiciones de exclusión mediante el uso de conjuntos dinámicos

Resolución de problemas MIP

- OPTCA=criterio_abs_optimalidad en MIP
- OPTCR=criterio_rel_optimalidad en MIP
- nombre_modelo.PRIOROPT = 1
- nombre_var.PRIOR = número
- Las variables más importantes deben ser las primeras en ramificar (mayor prioridad, i.e., número más bajo)
- SOS1 (como mucho una variable es diferente de 0 en un conjunto), SOS2 (como mucho dos variables son diferentes de 0 en un conjunto y deben ser adyacentes)

Detección de infactibilidades

- Parámetros de CPLEX
  - Anular el preproceso
    ```
    preind 0
    ```
  - Detección del conjunto mínimo de infactibilidades
    ```
    iis yes
    ```

Análisis de sensibilidad

- Parámetros de CPLEX
  - En coeficientes de función objetivo

    ```
    objrng all
    ```
  - En cotas de restricciones

    ```
    rhsrng all
    ```

Mejoras en la formulación

- reformulación manual del problema (especialmente indicado en problemas MIP)
- no crear variables ni ecuaciones superfluas
- reducción de número de restricciones y/o elementos
- escalado alrededor de 1 (especialmente indicado en problemas NLP)
- partir de un punto inicial (especialmente indicado en problemas NLP)
- acotamiento de variables

Cálculo analítico del número de restricciones

- Permite conocer a priori tamaños de problemas en función de parámetros del sistema
- Indica dónde se puede mejorar el modelado sin gran incremento de tamaño
- Permite comprobar la formulación matemática del problema

RESTRICCIONES	$2T + P(2B + C + H + 1) + SP(3T + 1) + NSP(D + KL + T)$
VARIABLES	$-C - H + P(C + H + 2T) + SP(T - 1) + NSP(2B + 3D + 2H + KL + T - 1)$

Preproceso

- Reducción de los tamaños de dos problemas con la opción de preproceso de CPLEX 6.0

	R	V	E	R	V	E
Sin presolve	19047	27262	81215	48971	63935	187059
Con presolve	15744	21982	51079	40794	56133	135361
Decremento	17,3%	19,4%	37,1%	16,7%	12,2%	27,6%

- Comparación entre preprocesos

	R	V	E	R	V	E
Sin preproc	19047	27847	82295	49715	64679	189477
Preproc CPLEX	-14,8%	-19,3%	-36,2%	-17,9%	-13,2%	-28,6%
Preproc OSL	-4,9%	0,0%	-2,4%	-15,6%	0,0%	-9,1%

Estructura de la matriz de restricciones y escalado

- Parámetros de GAMSCHK

  `advisory` identifica posibles variables y ecuaciones no acotadas o infactibles

  `analysis` analiza la estructura de la matriz de restricciones

  `blockpic` dibuja la matriz de restricciones por bloques

  `blocklist` dibuja el tamaño y características de cada bloque

- Parámetros de CPLEX
  - Calidad numérica de la solución

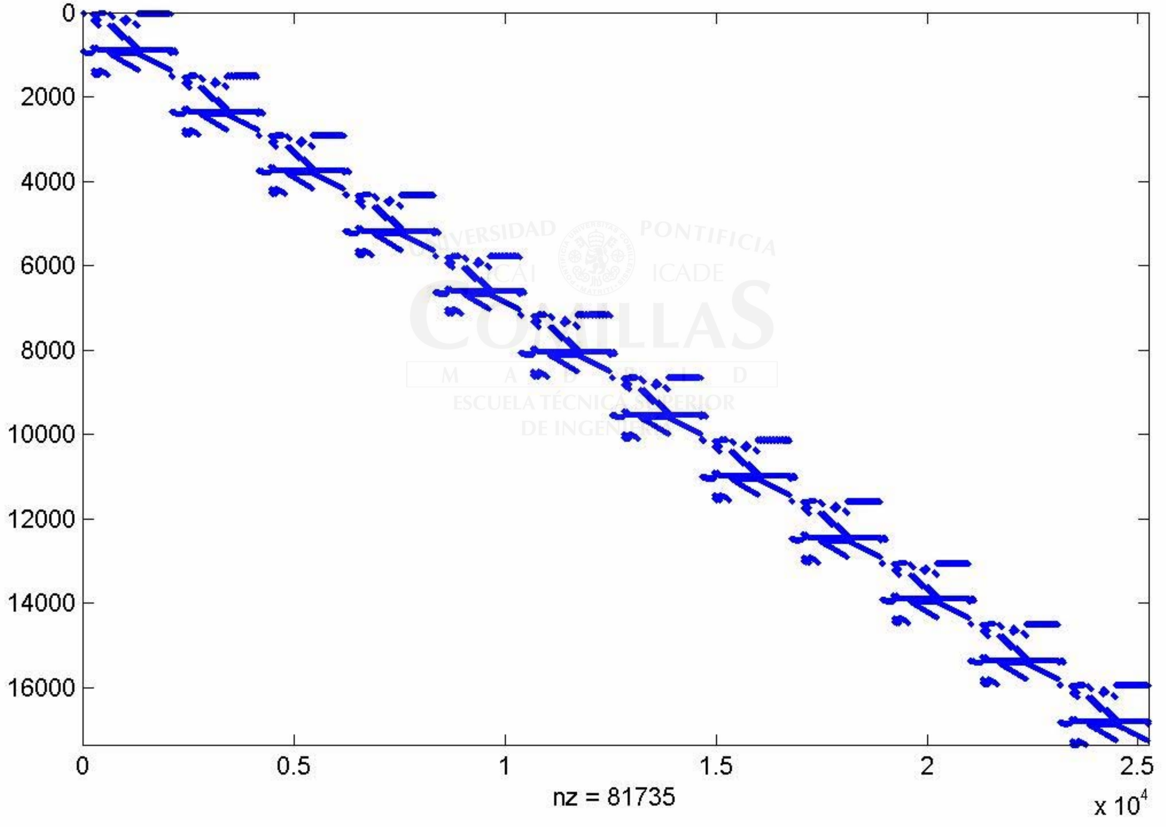
    `quality yes`
  - Escalado

    `scaind 0`

- Opciones de escalado en GAMS
  - nombre_modelo.`SCALEOPT` = 1 ;
  - nombre_var.`SCALE`=número ;
  - nombre_ec.`SCALE`=número ;

Estructura de la matriz de restricciones

- Ciertos algoritmos aprovechan la estructura del problema, i.e., técnicas de descomposición



Récords actuales en LP

- problema lineal estocástico de 1.100.000 x 1.600.000 x 4.400.000 resuelto en 1.5 h
- problema lineal de 117.000 x 225.000 x 655.000 resuelto en 240 s
- problema de 20.000 x 25.000 x 80.000 resuelto numerosas veces en un tiempo medio de 3 s
- problema de 50.000 x 65.000 x 200.000 resuelto numerosas veces en un tiempo medio de 8 s

Tiempos para procesador Pentium II a 233 MHz

Modelado en GAMS
Ejemplo de sistemas de energía eléctrica
Flujo óptimo de cargas en continua
DC-OPF

Andrés Ramos
Álvaro Baíllo

Índice

- El problema del flujo de cargas.
- El problema del flujo de cargas óptimo (OPF).
- Formulación en lenguaje GAMS.
- Resolución del problema.
- Análisis de resultados.

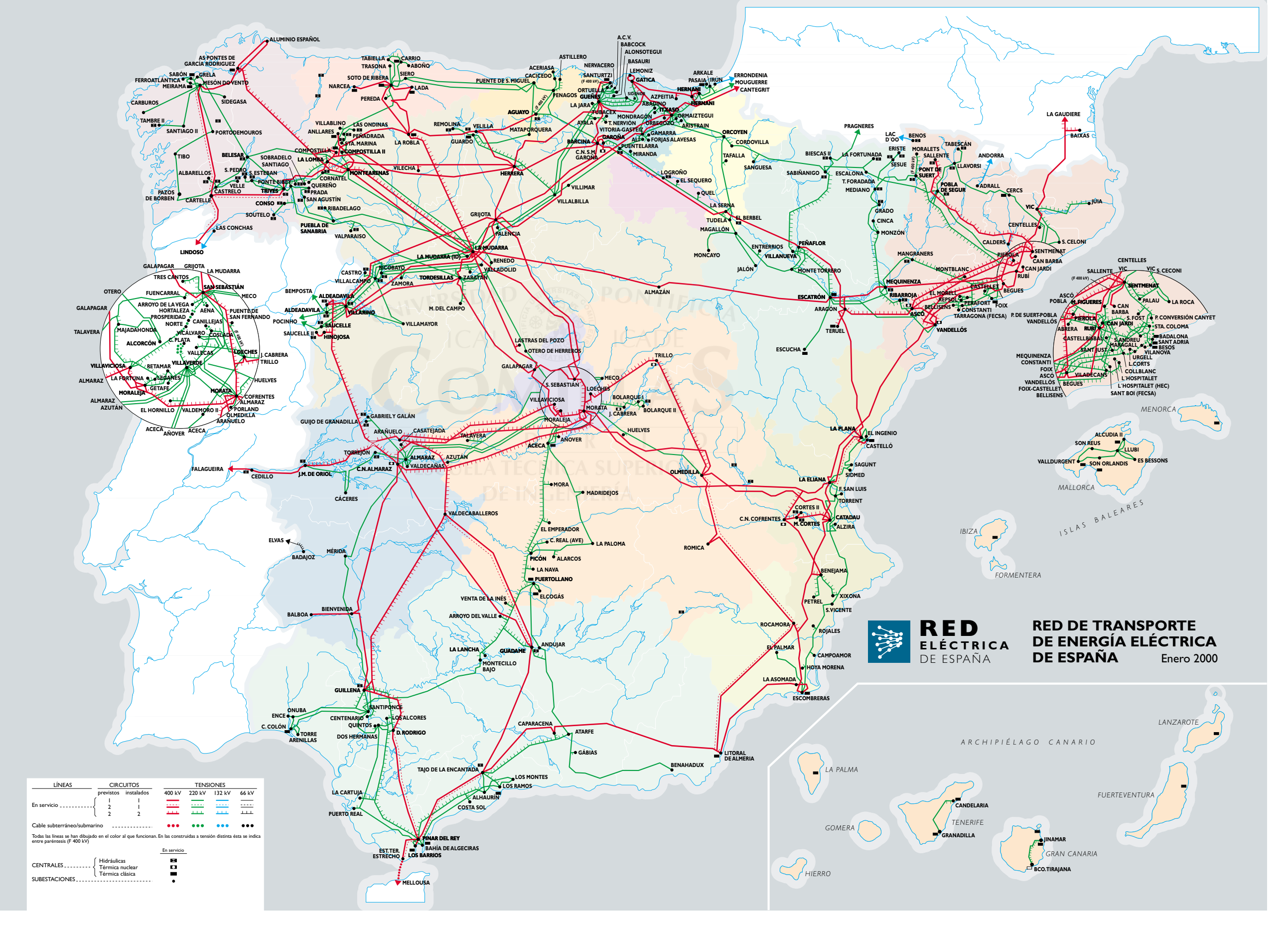
El problema del flujo de cargas (I)[1]

- Un sistema eléctrico interconectado está constituido por:
  - Una red de transporte de energía, formada por:
    - Ramas: Líneas y transformadores.
    - Nudos: Barras de subestaciones (buses).
  - Centrales de generación de energía, situadas en ciertos nudos (subestaciones de generación).
  - Centros de consumo de energía, situados en otros nudos (subestaciones de distribución).
- Cuestión fundamental: La energía eléctrica debe producirse en el mismo instante en que es consumida (no es almacenable de forma económica a gran escala).

[1]Elgerd, O.E. “Electric Energy Systems Theory: An Introduction.” Mc-Graw Hill Series in Electrical Engineering, 1983, pp. 219-273

El problema del flujo de cargas (II)

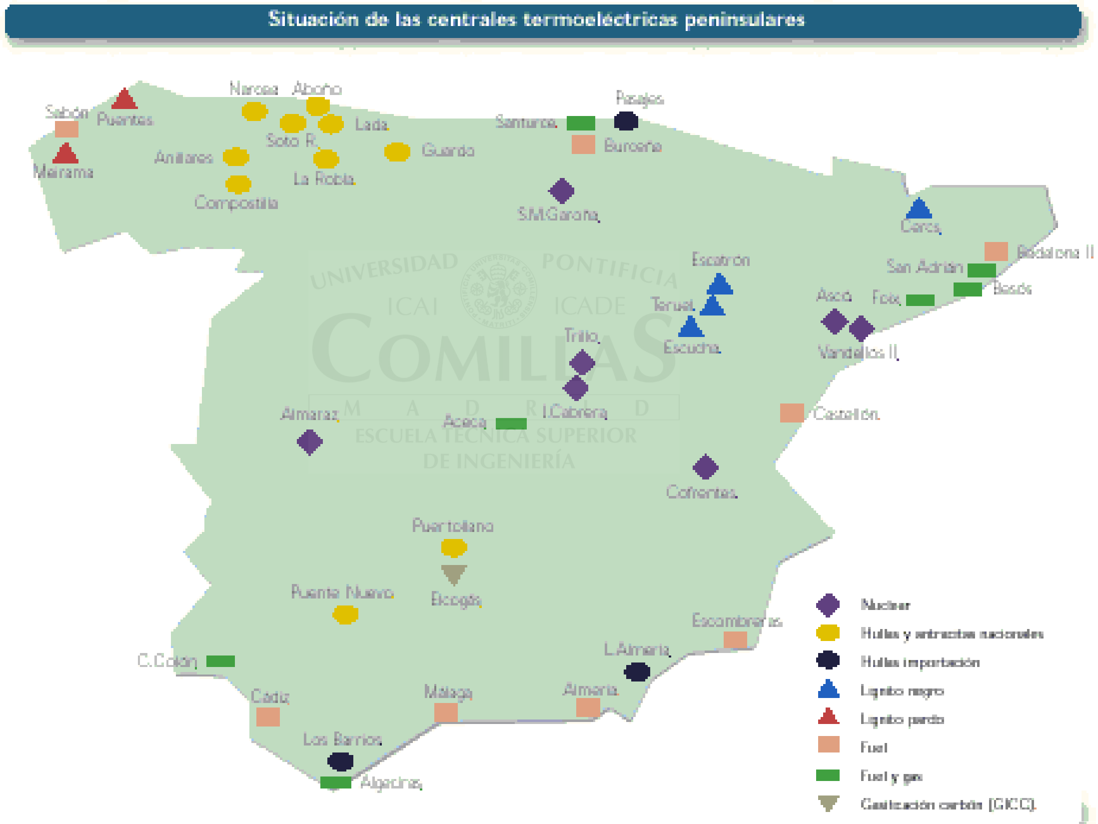
- Ejemplo de red de transporte[2]



[2]Red Eléctrica de España. Operación del Sistema Eléctrico. Informe 1999. Disponible en www.ree.es

El problema del flujo de cargas (III)

- Situación de las centrales de generación[2]



[2]Red Eléctrica de España. Operación del Sistema Eléctrico. Informe 1999. Disponible en www.ree.es

El problema del flujo de cargas (IV)

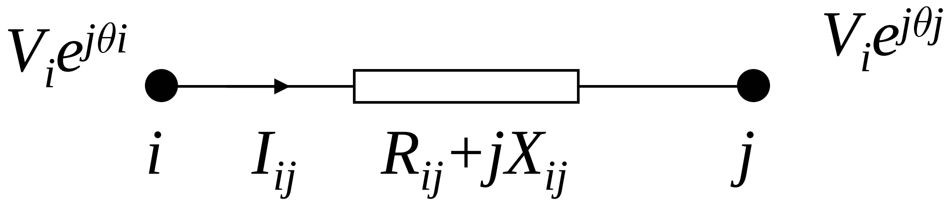
- Entre esos los centros de generación y los de consumo, la energía fluye por las líneas y centros de transformación de acuerdo con las leyes de Kirchoff.
- Es necesario vigilar esos flujos de potencia en **tiempo real**:
  - Los elementos de la red tienen unos límites de funcionamiento que no deben ser rebasados:
    - Límites térmicos de las líneas.
    - Límites de tensiones de los nudos.
  - El sistema de transporte de estudio puede estar interconectado con otros y existir un contrato de intercambio de potencia que hay que mantener (e.g. España con Francia o con Marruecos).
  - Un adecuado control del flujo de potencias permite evitar que el fallo de algún elemento tenga consecuencias desastrosas.

El problema del flujo de cargas (V)

- El análisis del flujo de cargas también se usa en **planificación**:
  - La empresa operadora de la red puede:
    - Planificar el mantenimiento de la red de transporte.
    - Planificar la expansión del sistema de transporte.
  - Las empresas de generación pueden:
    - Decidir el emplazamiento de nuevas instalaciones de generación (influyen otros factores como el fácil acceso al suministro de combustible).
- La configuración de la red de transporte provoca **diferencias en el precio de la energía eléctrica** entre los distintos nudos de la red debido a las **pérdidas** o las **congestiones**.

El problema del flujo de cargas (VI)

- Ecuaciones del flujo de cargas:



$$S_{ij} = V_i e^{j\theta i} \vec{I}_{ij}^{\,*} =$$

$$= V_i\left(\cos\theta_i + j\sin\theta_i\right)\frac{V_i\left(\cos\theta_i - j\sin\theta_i\right) - V_j\left(\cos\theta_j - j\sin\theta_j\right)}{R_{ij} - jX_{ij}}$$

- Operando se llega a:

$$P_{ij} = \frac{1}{R_{ij}^{\ 2} + X_{ij}^{\ 2}}\left(X_{ij}V_iV_j\sin\left(\theta_i - \theta_j\right) + R_{ij}\left(V_i^2 - V_iV_j\cos\left(\theta_i - \theta_j\right)\right)\right)$$

$$Q_{ij} = \frac{1}{R_{ij}^{\ 2} + X_{ij}^{\ 2}}\left(X_{ij}\left(V_i^2 - V_iV_j\cos\left(\theta_i - \theta_j\right)\right) - R_{ij}V_iV_j\sin\left(\theta_i - \theta_j\right)\right)$$

El problema del flujo de cargas (VII)

- Las pérdidas en la línea vienen dadas por la suma de la potencia que sale de *i* hacia *j* y la potencia que sale de *j* hacia *i*:

$$P_{ij} + P_{ji} = \frac{R_{ij}}{R_{ij}^{\ 2} + X_{ij}^{\ 2}} \left(V_i^2 + V_i^2 - 2V_iV_j \cos\left(\theta_i - \theta_j\right) \right)$$

- El flujo de cargas consiste en un sistema de ecuaciones no lineal.
- La resolución del problema del flujo de cargas se lleva a cabo mediante sofisticadas herramientas informáticas (e.g. PSS/E).

El problema del flujo de cargas (VII)

- Variables incontroladas: escapan al control de la empresa operadora de red:
  - Potencias activa y reactiva en los nudos de consumo.
- Variables de control: pueden ser controladas:
  - Potencias activa y reactiva en los nudos de generación
- Variables de estado: describen el sistema:
  - Módulos y argumentos de las tensiones de nudo.
- Cuatro tipos de nudo:

	$\|V\|$	θ	P	Q
– Nudo de generación:	*		*	
– Nudo de consumo:			*	*
– Nudo de referencia:		*	*	(*)
– Nudo balance:	*			

*: Conocido

El problema del flujo de cargas (VIII)

- Simplificaciones (flujo de cargas DC):
  - $V_i = V_j = V$
  - $X_{ij} >> R_{ij}$
  - $\theta_i \simeq \theta_j \Rightarrow \begin{cases} \cos(\theta_i - \theta_j) \simeq 1 \\ \sin(\theta_i - \theta_j) \simeq \theta_i - \theta_j \end{cases}$

- Flujo de activa: $$P_{ij} = \frac{V^2(\theta_i - \theta_j)}{X_{ij}} = S_B \frac{\theta_i - \theta_j}{x_{ij}}$$

- Pérdidas: $$P_{ij} + P_{ji} = \frac{2R_{ij}}{R_{ij}^2 + X_{ij}^2} V^2 \left(1 - \cos(\theta_i - \theta_j)\right)$$

$$= S_B \frac{2r_{ij}}{r_{ij}^2 + x_{ij}^2} \left(1 - \cos(\theta_i - \theta_j)\right)$$

El problema del flujo de cargas óptimo (I)

- El problema del flujo de cargas considera como dato la potencia activa producida en los nudos de generación.
- Sin embargo, en la planificación de la generación para el corto plazo, puede ser interesante decidir la producción de cada generador teniendo en cuenta:
  - Costes variables de producción de los generadores.
  - Límites físicos de los elementos de la red de transporte.
  - Pérdidas.
- El problema de decidir la explotación de la generación a corto plazo considerando la red de transporte se denomina flujo de cargas óptimo (**Optimal Power Flow**)
- Vamos a plantear un **OPF** que represente la red de transporte mediante la simplificación **DC**.

El problema del flujo de cargas óptimo (II)

- Se trata de **minimizar los costes variables de operación** en un intervalo horario:
  - costes variables de los grupos térmicos
  - costes de oportunidad de los grupos hidráulicos cuando producen por encima de su potencia programada.
  - coste variable de la potencia no suministrada.

$$\sum_{t=1}^{T} v_t GTR_t + \sum_{h=1}^{H} v_h GHE_h + \sum_{n=1}^{N} v_n PNS_n$$

  - Datos: v_t: coste variable de generación del grupo térmico t.

    v_h: coste de oportunidad de la hidráulica de emergencia.

    v_h: coste variable de la potencia no suministrada.

  - Variables: GTR_t potencia producida por el grupo térmico t.

    GHE_h potencia hidráulica de emergencia del grupo h.

    PNS_n potencia no suministrada en el nudo n.

El problema del flujo de cargas óptimo (III)

- Cotas de las variables del equipo generador:
  - Potencia térmica máxima y mínima del grupo *t*:

$$\underline{GTR_t} \le GTR_t \le \overline{GTR_t}$$

  - La potencia hidráulica programada máxima del grupo *h*:

$$0 \le GHP_h \le \overline{GHP_h}$$

  - La potencia hidráulica de emergencia máxima del grupo *h*:

$$0 \le GHE_h \le \left(\overline{GHM_h} - \overline{GHP_h}\right)$$

  - La potencia no suministrada como mucho será la demanda del nudo.

$$0 \le PNS_n \le D_n$$

El problema del flujo de cargas óptimo (IV)

- Modelado de la red de transporte:
  - 1ª Ley Kirchoff: Balance entre generación y demanda de nudo:

$$\sum_{t\in n} GTR_t + \sum_{h\in n}(GHP_h + GHE_h) + PNS_n + \sum_{i=1}^{I} F_{i\to n} - \sum_{j=1}^{J} F_{n\to j} = D_n$$

  - 2ª Ley Kirchoff: Flujo de potencia activa por las líneas:

$$\frac{X_{i\to j}}{S_B} F_{i\to j} = \theta_i - \theta_j$$

  - Cotas de los flujos: $-\overline{F_{i\to j}} \le F_{i\to j} \le \overline{F_{i\to j}}$

El problema del flujo de cargas óptimo (V)

- Modelado de la red de transporte:
  - Formulación alternativa de la 1ª Ley Kirchoff:

$$\sum_{t \in n} GTR_t + \sum_{h \in n} (GHP_h + GHE_h) + PNS_n +$$

$$\sum_{i=1}^{I} (\theta_i - \theta_n) S_B / X_{i \to n} - \sum_{j=1}^{J} (\theta_n - \theta_j) S_B / X_{n \to j} = D_n$$

  - Límites térmicos de las líneas como restricciones:

$$\theta_i - \theta_j \leq \overline{F_{i \to j}} \frac{X_{i \to j}}{S_B}$$

$$\theta_i - \theta_j \geq -\overline{F_{i \to j}} \frac{X_{i \to j}}{S_B}$$

  - Esta formulación tiene menos variables, pero más restricciones.

El problema del flujo de cargas óptimo (VI)

- Si se consideran las pérdidas:
  - Las pérdidas óhmicas de una línea se modelan con una expresión *no lineal*:

$$L_{i\to j} = 2S_B \frac{r_{i\to j}}{r_{i\to j}^2 + X_{i\to j}^2}\left[1-\cos(\theta_{\mathrm{i}}-\theta_j)\right]$$

  - Las pérdidas se incluyen como dos cargas adicionales iguales en los extremos de la línea.
  - 1ª Ley Kirchhoff:

$$\sum_{t\in n} GTR_t + \sum_{h\in n}(GHP_h + GHE_h) + PNS_n + \sum_{i=1}^{I} F_{i\to n} - \sum_{j=1}^{J} F_{n\to j} = D_n + L_n$$

  - siendo las pérdidas en el nudo n:

$$L_n = \left(\sum_{i=1}^{I} L_{i\to n} + \sum_{j=1}^{J} L_{n\to j}\right)/2$$

Formulación del DC-OPF en GAMS (I)

```
$TITLE Flujo de cargas en corriente continua con y sin pérdidas

SETS
   ND           nudos
   GR           generadores
   TR(gr)       generadores térmicos
   HD(gr)       generadores hidráulicos
   NDGR(nd,gr) localización de generadores en nudos
   LN(nd,nd)    líneas

   CN características nudos       / dem, cpns /
   CG características generadores / coste, pmin, pmax, cshd, hdrpro, hdrmax /
   CL características líneas      / r, x, flmax /

ALIAS (nd, ni, nf) ;

SCALARS
   SBASE  potencia base [GW]                             / 0.1 /
   OPCPRD opción de modelado de las pérdidas (no 0 si 1) / 0 /

* definición de la estructura de datos sin incluir explícitamente éstos

PARAMETERS
   DATNUD(nd,cn)    datos de los nudos
   DATGEN(gr,cg)    datos de los generadores
   DATLIN(nd,nd,cl) datos de las líneas
```

Formulación del DC-OPF en GAMS (II)

```
* planteamiento matemático del problema

VARIABLES
   COSTE       función objetivo                              [MPTA]
   TT(nd)      ángulo de tensión en el nudo                  [rad]
   FL(ni,nf)   flujo de potencia                             [GW]

POSITIVE VARIABLES
   GTR(gr)     generación térmica                            [GW]
   GHP(gr)     generación hidráulica programada              [GW]
   GHE(gr)     generación hidráulica de emergencia           [GW]
   PNS(nd)     potencia no suministrada                      [GW]
   PRDAS(nd) pérdidas de las líneas conectadas al nudo [GW]

EQUATIONS
   FO          costes de generación y de indisponibilidad [MPTA]
   KR1F(nd)    primera ley de Kirchhoff para cada nudo en función de flujos
   KR1A(nd)    primera ley de Kirchhoff para cada nudo en función de ángulos
   FLJ(ni,nf)  flujo en función de ángulos de tensión
   FLJP(ni,nf) diferencia angular máxima en cada línea en un   sentido
   FLJN(ni,nf) diferencia angular máxima en cada línea en otro sentido
   EPRDAS(nd)  pérdidas de las líneas conectadas al nudo ;
```

Formulación del DC-OPF en GAMS (III)

```
FO      .. COSTE =E= SUM[tr, DATGEN(tr,'coste') * GTR(tr)]
                + SUM[hd, DATGEN(hd,'cshd')  * GHE(hd)]
                + SUM[nd, DATNUD(nd,'cpns')  * PNS(nd)] ;

KR1F(nd) ..  SUM[NDGR(nd,tr), GTR(tr)] + SUM[NDGR(nd,hd), GHP(hd) + GHE(hd)]
           + SUM[LN(ni,nd), FL(ni,nd)] - SUM[LN(nd,nf), FL(nd,nf)]
           + PNS(nd) =E= DATNUD(nd,'dem') + PRDAS(nd) $OPCPRD ;

KR1A(nd) .. SUM[NDGR(nd,tr), GTR(tr)]
          + SUM[NDGR(nd,hd), GHP(hd) + GHE(hd)]
          + SUM[LN(ni,nd), (TT(ni) - TT(nd)) / DATLIN(ni,nd,'x')] * SBASE
          - SUM[LN(nd,nf), (TT(nd) - TT(nf)) / DATLIN(nd,nf,'x')] * SBASE
          + PNS(nd) =E= DATNUD(nd,'dem') + PRDAS(nd) $OPCPRD ;

FLJ(LN(ni,nf)) .. FL(ni,nf) * DATLIN(ni,nf,'x') / SBASE =E= TT(ni) - TT(nf) ;

FLJP(LN(ni,nf)) ..
   TT(ni) - TT(nf) =L=   DATLIN(ni,nf,'flmax') * DATLIN(ni,nf,'x') / SBASE ;

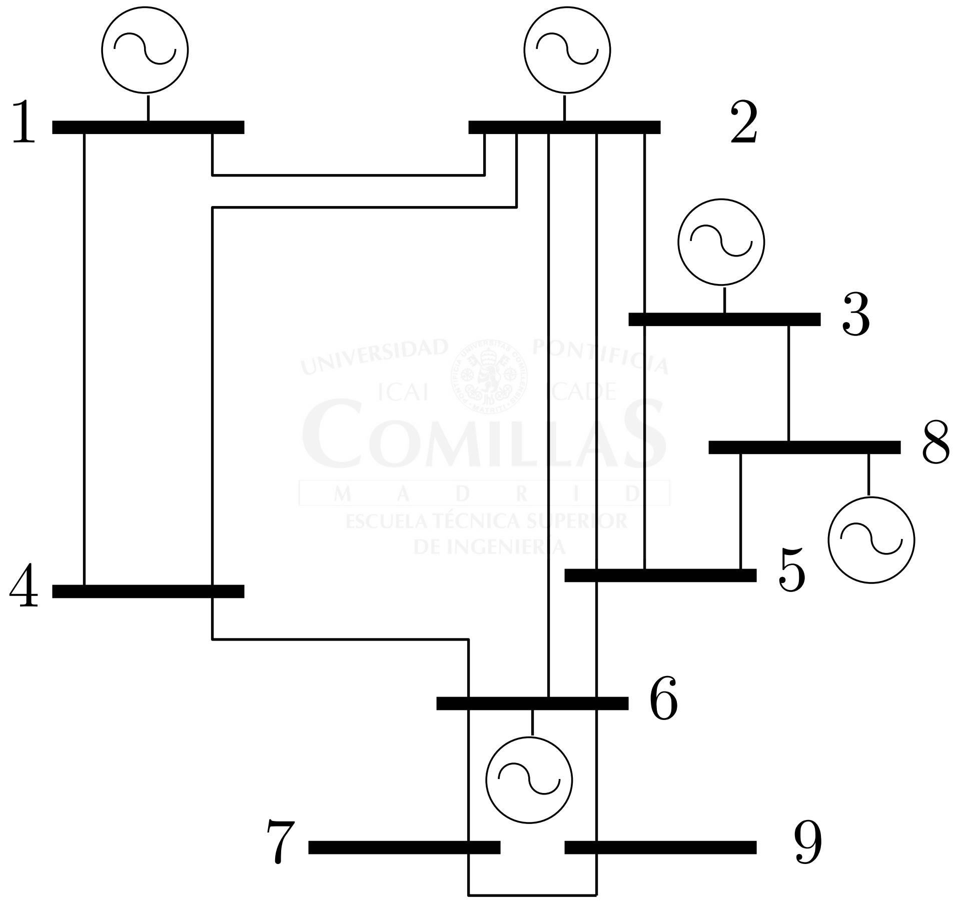
FLJN(LN(ni,nf)) ..
   TT(ni) - TT(nf) =G= - DATLIN(ni,nf,'flmax') * DATLIN(ni,nf,'x') / SBASE ;

EPRDAS(nd) .. PRDAS(nd) =E= SBASE * SUM[LN(ni,nd), (1-cos(TT(ni) - TT(nd)))*
              DATLIN(ni,nd,'r')/(DATLIN(ni,nd,'r')**2+DATLIN(ni,nd,'x')**2)]
            + SBASE * SUM[LN(nd,nf), (1-cos(TT(nd) - TT(nf)))*
              DATLIN(nd,nf,'r')/(DATLIN(nd,nf,'r')**2+DATLIN(nd,nf,'x')**2)] ;
```

Formulación del DC-OPF en GAMS (IV)

```
MODEL FC  / FO, KR1F, FLJ         / ;
MODEL FCA / FO, KR1A, FLJP, FLJN  / ;
MODEL FCP / FO, KR1F, FLJ, EPRDAS / ;
```

Formulación del DC-OPF en GAMS (V)



Formulación del DC-OPF en GAMS (VI)

```
* caso de estudio
*** esta parte iría en ficheros independientes y se introduciría con $include

SETS
   ND     nudos                        / nudo-1 * nudo-9 /
   GR     generadores                  / genr-1 * genr-9, genh-1 * genh-4 /
   NDGR(nd,gr) localización de generadores en nudos
  /
   nudo-1 . genr-1
   nudo-1 . genr-2
   nudo-1 . genr-3
   nudo-2 . genr-4
   nudo-2 . genr-5
   nudo-2 . genr-6
   nudo-3 . genr-7
   nudo-3 . genr-8
   nudo-3 . genr-9
   nudo-1 . genh-1
   nudo-3 . genh-2
   nudo-6 . genh-3
   nudo-8 . genh-4
  / ;
```

Formulación del DC-OPF en GAMS (VII)

```
TABLE DATNUD(nd,cn) datos de los nudos
             dem   cpns
*             MW PTA/kWh
   nudo-1      1   150
   nudo-2    240   150
   nudo-3     40   150
   nudo-4    160   150
   nudo-5    240   150
   nudo-6     80   150
   nudo-7    100   150
   nudo-8     15   150
   nudo-9    100   150 ;
```

Formulación del DC-OPF en GAMS (VIII)

```
TABLE DATGEN(gr,cg) datos de los generadores
            coste   pmin  pmax   cshd  hdrpro hdrmax
*          PTA/kWh   MW    MW  PTA/kWh   MW     MW
   genr-1    6.5      0    75
   genr-2    7.0      0   125
   genr-3    7.5      0   100
   genr-4    5.9      0   100
   genr-5    6.7      0    50
   genr-6    7.4      0    50
   genr-7    6.1      0   100
   genr-8    7.6      0    50
   genr-9    8.0      0    50
   genh-1                         1.0   300     300
   genh-2                         1.0   160     160
   genh-3                         1.0   150     150
   genh-4                         1.0   100     100 ;
```

Formulación del DC-OPF en GAMS (IX)

```
TABLE DATLIN(ni,nf,cl) datos de las líneas
                    r       x       flmax
*                   p.u.    p.u.      MW
   nudo-1 . nudo-2  0.0777  0.2913  500
   nudo-1 . nudo-4  0.0544  0.2041  500
   nudo-2 . nudo-3  0.0424  0.1695  500
   nudo-2 . nudo-4  0.1     0.4     500
   nudo-2 . nudo-5  0.05    0.2     500
   nudo-2 . nudo-6  0.1     0.4     500
   nudo-3 . nudo-5  0.0248  0.099   500
   nudo-3 . nudo-8  0.1     0.4     500
   nudo-4 . nudo-6  0.15    0.6     500
   nudo-5 . nudo-6  0.05    0.2     500
   nudo-5 . nudo-8  0.1     0.4     500
   nudo-6 . nudo-7  0.15    0.6     500
   nudo-6 . nudo-9  0.05    0.2     500
   nudo-7 . nudo-9  0.05    0.2     500 ;

*** hasta aquí son ficheros independientes
```

Formulación del DC-OPF en GAMS (X)

```
* activación de generadores térmicos hidráulicos y líneas

TR(gr)    $DATGEN(gr,'pmax')   = YES ;
HD(gr)    $DATGEN(gr,'hdrpro') = YES ;
LN(ni,nf) $DATLIN(ni,nf,'x')   = YES ;




* escalación de datos de potencia a GW

DATNUD(nd,'dem')    = DATNUD(nd,'dem')    / 1e3 ;
DATGEN(tr,'pmin')   = DATGEN(tr,'pmin')   / 1e3 ;
DATGEN(tr,'pmax')   = DATGEN(tr,'pmax')   / 1e3 ;
DATGEN(hd,'hdrpro') = DATGEN(hd,'hdrpro') / 1e3 ;
DATGEN(hd,'hdrmax') = DATGEN(hd,'hdrmax') / 1e3 ;
DATLIN(ln,'flmax')  = DATLIN(ln,'flmax')  / 1e3 ;
```

Formulación del DC-OPF en GAMS (XI)

```
* acotamiento de las variables (cotas físicas)

GTR.LO(tr) = DATGEN(tr,'pmin') ;
GTR.UP(tr) = DATGEN(tr,'pmax') ;

GHP.UP(hd) = DATGEN(hd,'hdrpro') ;
GHE.UP(hd) = DATGEN(hd,'hdrmax') - DATGEN(hd,'hdrpro') ;

PNS.UP(nd) = DATNUD(nd,'dem') ;

FL.LO(ln)  = - DATLIN(ln,'flmax') ;
FL.UP(ln)  =   DATLIN(ln,'flmax') ;

* cotas algorítmicas de los ángulos

TT.LO(nd) = - 1.5 ;
TT.UP(nd) =   1.5 ;

* nudo de referencia

TT.FX(nd) $(ORD(nd) EQ 1) = 0 ;
```

Formulación del DC-OPF en GAMS (XII)

```
* opción sin pérdidas
OPCPRD = 0 ;
* flujo de cargas con variables de flujo
SOLVE FC  USING  LP MINIMIZING COSTE ;
* control sobre aprovechamiento de base previa
OPTION BRATIO = 1 ;
* flujo de cargas con variables de ángulos de tensión
SOLVE FCA USING  LP MINIMIZING COSTE ;
* opción con pérdidas
OPCPRD = 1 ;
* flujo de cargas con variables de flujo
SOLVE FCP USING NLP MINIMIZING COSTE ;
```