

MEMORIA JUSTIFICATIVA

En las siguientes páginas se implementa un algoritmo genético (AG) para la resolución del problema de planificación en sistemas eléctricos de potencia consistente en: "Determinar la distribución de condensadores que optimiza la rentabilidad económica de la inversión necesaria para minimizar los costes de explotación del sistema".

Se contempla un horizonte temporal en el que se consideran unos estados típicos del sistema representativos de la evolución diaria y estacional de la demanda (escenarios temporales).

Una distribución dada de condensadores comprende tanto la ubicación, como la potencia (número de escalonamientos) y el tipo de los bancos que la integran.

El AG es un algoritmo de búsqueda que emplea técnicas basadas en la selección natural y en la genética, operando en dos fases:

1ª.- Comienza por crear un conjunto inicial de distribuciones válidas para constituir lo que se denomina una población inicial (POB).

2ª.- Mediante manipulaciones sobre la población inicial, que incluyen la selección de parejas de distribuciones (en función del grado de bondad que ofrezcan como soluciones del problema), combinación y mutación de sus propiedades, el AG genera una nueva población (NPOB), que es "hija" de la anterior y "padre" para la siguiente población que se obtenga por el mismo procedimiento.

En definitiva, el AG reproduce un proceso evolutivo encaminado a obtener cada vez mejores soluciones.

7. Arquitectura del AG

Son dos los objetivos que se persiguen:

- 1°.- Minimizar el coste de las pérdidas eléctricas del sistema.
- 2°.- Maximizar la rentabilidad de la inversión (supeditado al primer objetivo).

Dado que se contemplan varios escenarios temporales y que se permiten bancos de condensadores regulables, la estrategia que se sigue en el diseño del AG consiste en lo siguiente:

- 1.- Clasificar los escenarios según las pérdidas eléctricas que presenten. Se designa como CASO PRINCIPAL a aquel escenario que ocupe el primer lugar en el ranking por pérdidas. El resto de los escenarios se denominan CASOS SECUNDARIOS.
- 2.- Construir un algoritmo genético, Algoritmo Genético Principal (AGP), que efectúe un proceso evolutivo con distribuciones "principales" (verificadas sobre el caso principal) con el objetivo de maximizar el valor actual neto (VAN) de la inversión (abreviadamente, problema "max VAN").
3. Emplear un segundo algoritmo genético, denominado Algoritmo Genético Básico (AGB), que ajuste los escalones de potencia de los bancos que componen cada distribución principal con el objetivo de minimizar las pérdidas en los casos secundarios. Por decirlo así, el AGB trabaja con distribuciones derivadas (secundarias) de una principal, suponiendo en principio que todos los bancos son regulables. Comparando las distribuciones principal y secundarias, se deduce finalmente de qué tipos son los bancos. Determinados las potencias y los tipos de los condensadores, es posible calcular la inversión asociada. Por otro lado, el conocimiento de las pérdidas eléctricas conduce

inmediatamente a la apreciación de los costes de explotación.

El VAN se obtiene de la expresión:

$$VAN(X, ta, vu) = -D(X) + N(X) \cdot \frac{(r - r^{vu+1})}{1 - r}$$

Con:
$$r = \frac{1}{1 + ta}$$

Donde:

- D = Desembolso inicial de la inversión (€).
- N = Flujo neto de caja anual (€).
- ta = Tasa de actualización que se estima como coste de capital o como rendimiento apropiado.
- vu = Vida útil de la inversión (años).

El flujo de caja neto anual se considera constante para toda la vida útil de la inversión, y se calcula como la suma, extendida a todos los escenarios, de los decrementos habidos en las pérdidas eléctricas por el tiempo y por el precio del Kilovatio-hora en el mercado eléctrico.

Dado que el término que entra en la ecuación del VAN es el flujo de caja y no las pérdidas eléctricas, y dado que minimizar las pérdidas eléctricas es equivalente a maximizar los flujos de caja, este último criterio es el que se adopta en el AGB (abreviadamente, "max N").

En la figura 7.1 se muestra una representación global de la arquitectura propuesta para el AG.

En primer lugar se tiene un módulo denominado DATOS. Toda la información de entrada al AG se lee de varios ficheros y se almacena en las variables que serán descritas en el capítulo 8.

En el módulo INICIO, se inician las variables internas del algoritmo y se definen las estructuras "individuo" empleadas por el algoritmo principal (AGP). Cada individuo es una posible solución del problema "max VAN", y un conjunto de ellos conforma una población. En cada

iteración, se tendrá una población POBP constituida por los padres y NPOBP, correspondiente a los hijos. Un individuo se compone a su vez de una serie de campos: Genotipo, Fenotipo, Pérdidas, Estado y Rentabilidad (Véase capítulo 9). Los cuatro campos primeros están definidos para la distribución principal y las secundarias resultantes (proporcionadas por el AGB).

En los dos siguientes bloques se implementan las dos fases comentadas de un algoritmo genético. Se trata de los módulos POB_INICIAL Y EVOLUCION del AGP (capítulos 10 y 11). Tanto en la creación de la población inicial como en las operaciones que se efectúan en el proceso evolutivo, el algoritmo trabaja con la codificación binaria (genotipo de las estructuras individuo). Por tanto, es preciso que en ambos se ejecute una rutina llamada COMPLETA_IND, encargada de completar la estructura individuo, es decir, asignar los valores correspondientes a los campos fenotipo, pérdidas, estado y rentabilidad. Tal como se ha expuesto anteriormente, para encontrar las distribuciones derivadas, soluciones del problema "max N", se emplea un algoritmo genético interior: el AGB (capítulo 13). El módulo COMPLETA_IND pone en marcha el AGB tantas veces como escenarios secundarios se disponga.

En la figura 7.1. se representa con cuatro cajas los submódulos de que consta el AGB. De nuevo se observa que presenta un módulo inicial, AGB_INICIO, y sendos módulos AGB_POB_INICIAL y AGB_EVOLUCION, con las mismos cometidos que sus homólogos en el AGP. En AGB_INICIO se define una nueva estructura "individuo", sobre la que se construyen las poblaciones secundarias POBS y NPOBS (padre e hija, respectivamente). Esta vez los campos de la estructura están definidos para una sola distribución, y el Flujo de Caja sustituye a la Rentabilidad.

En ambos algoritmos genéticos, se implementa unas rutinas gráficas que permiten presentar informes por pantalla con la siguiente

periodicidad:

1. Generación tras generación (módulos INFOR_GENERACIÓN y AGB_INFOR_GENERACIÓN): Composición poblacional (capítulo 10).
2. Al finalizar sus respectivas actuaciones (módulos RESULTADOS y AGB_RESULTADOS): Gráficas de la evolución y mejores soluciones obtenidas (capítulo 12).

Los resultados finales se vuelcan en un fichero de salida (solucion.txt).

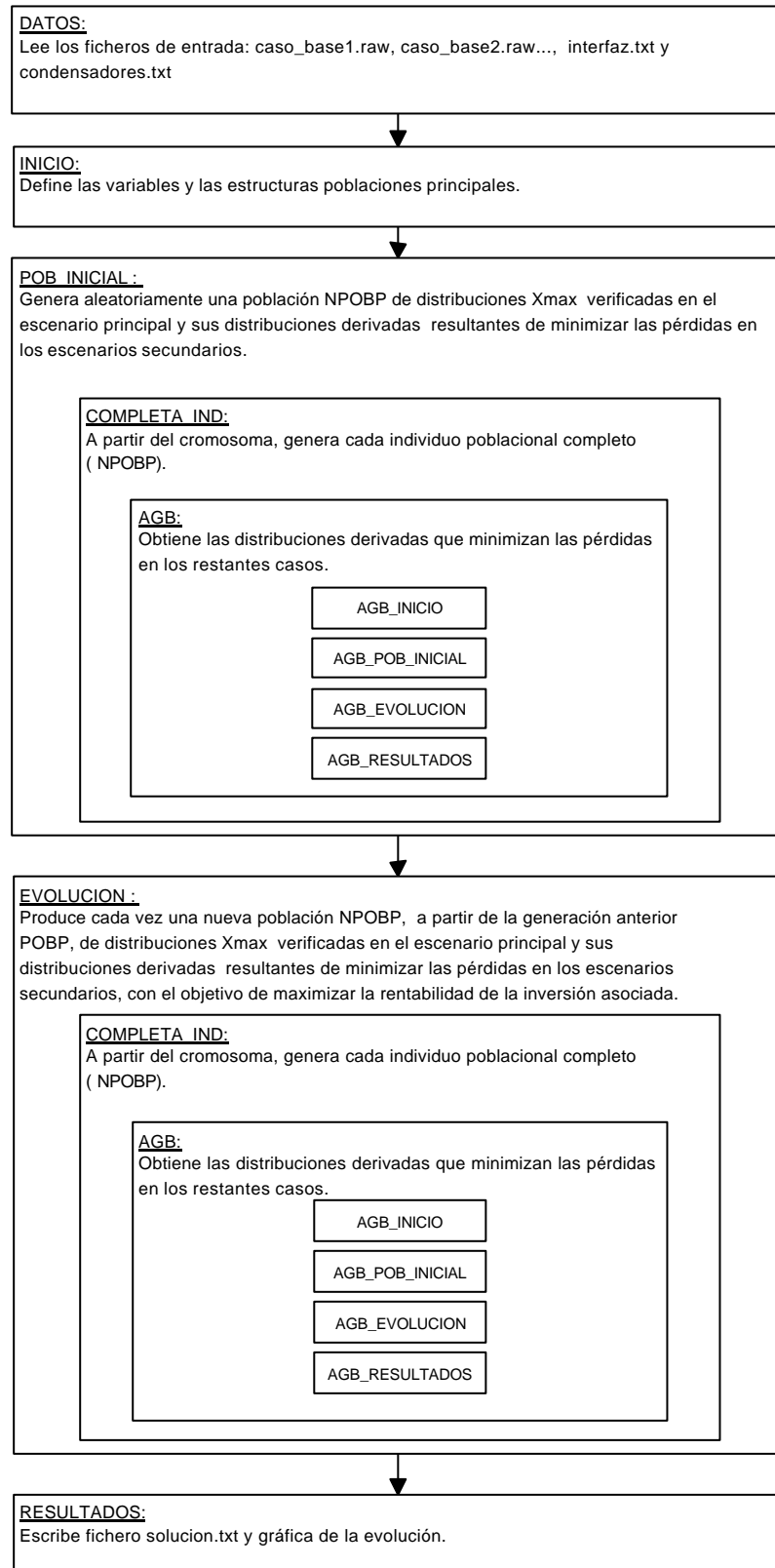


Figura 7 1 Algoritmo Genético Principal (AGP)

8. Módulo DATOS

En este módulo, se lee de varios ficheros la información necesaria para el funcionamiento del programa y se almacena en las variables que se describen en este capítulo. En la figura 8.1 se muestra el diagrama de flujo.

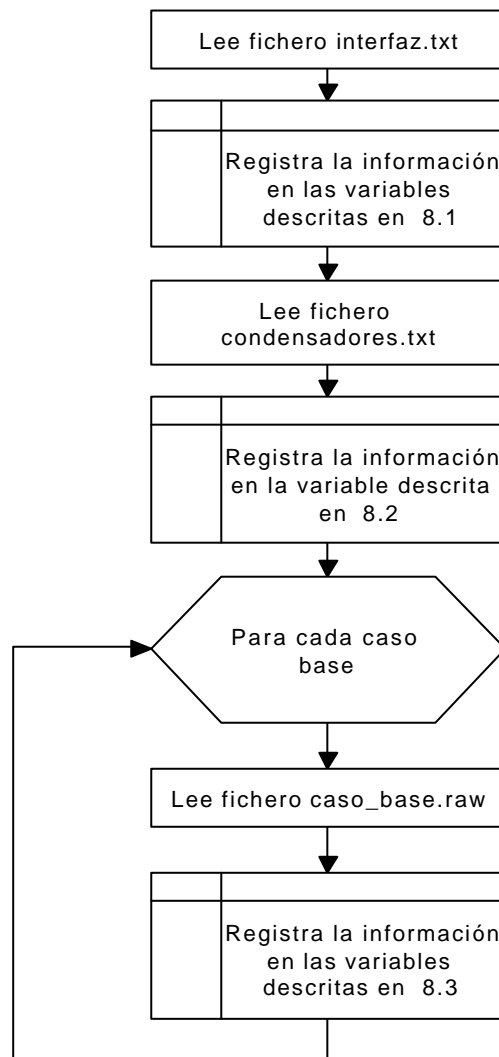


Figura 8.1. Módulo DATOS.

Llama a las tres funciones que se explican a continuación.

8.1. Función LEEINTERFAZ

Lee el fichero interfaz.txt. Recibe este nombre porque está dedicado a servir de enlace entre el usuario y el programa.

La función LEEINTERFAZ tiene la siguiente sintaxis:

```
[casos_base,ha,ema,vlimPQ,vlimPV,vu,ta,pKWh,nc,num_pasos,KV_T,num_ind,num_gen,pcr,pmu,nm,np]= leeinterfaz(nombre_fichero)
```

8.1.1. Variable de entrada

nombre_fichero: 'interfaz.txt' o cualquier otro fichero con idéntica estructura.

```
* FICHERO INTERFAZ.TXT
*
***Nombres de los ficheros .raw y horas/año de vigencia
TS1S2.raw 1000
TS1S1.raw 6760
TS1S0.raw 1000
*** Error máximo admisible para el reparto de carga(potencia en p.u.)
0.001
*** Tensiones límites(en p.u.): vsup_PQ,vinf_PQ,vsup_PV,vinf_PV
1.05 0.85 1.02 0.98
*** Vida útil de la inversión(años), tasa de actualización y precio del
Kilovatio-hora(euros).
10 0.05 0.038
*** Número de nudos candidatos y número máximo de pasos para los bancos
6 7
*** Para cada nivel de tensión(KV), clase(tamaño) de
condensador(1,2,3...)
23 1
0
*** Parámetros de funcionamiento del
AG:num_ind,num_gen,pcr,pmu,nm,np,elite,umbral_div
20 10 0.98 0.01 5 1 1 15
```

Figura 8.2. Ejemplo de fichero interfaz.txt.

8.1.2. Variables de salida

El fichero interfaz.txt contiene una serie de variables que pueden ser modificadas y para las que el AG encontrará distintas soluciones.

8.1.2.1. Nombre de los casos base y horas/año de vigencia

Junto al nombre del fichero *.raw , aparece en el fichero el número de horas al año que el escenario correspondiente es representativo del estado del sistema eléctrico. En el ejemplo (figura 8.2), la situación modelada por TS1S1 se produce durante 6760 horas al año. Se almacenan en las variables:

casos_base: Matriz de caracteres compuesta por filas que son los nombres de los ficheros caso_base.raw.

ha: Vector de horas/año de validez de los escenarios contemplados.

Para el ejemplo:

$$\text{casos_base} = \begin{bmatrix} TS1S2 \\ TS1S1 \\ TS1S0 \end{bmatrix} \quad \text{ha} = \begin{bmatrix} 1000 \\ 6760 \\ 1000 \end{bmatrix}$$

8.1.2.2. Error máximo admisible para el reparto de carga

El cálculo de las pérdidas eléctricas procede de la estimación del estado del sistema a través de un proceso iterativo basado en el balance de potencias en los nudos (reparto de carga). La convergencia de este algoritmo se alcanza cuando el error es inferior a un valor umbral que establece el grado de exactitud de las soluciones que obtenga.

Este valor se guarda en:

ema: Error máximo admisible para el reparto de carga (potencia en p.u.). Para el ejemplo, *ema*= 0.001 p.u.

8.1.2.3. Tensiones admisibles

Para los estados del sistema provocados por las distribuciones de condensadores, se considera un rango de tensiones límite en los nudos de consumo y otro en los nudos de generación. Las variables correspondientes en el programa son:

vlimPQ: Vector columna cuyo primer elemento, *vlimPQ(1)*, contiene el límite superior del módulo de la tensión (en p.u.) para los nudos PQ, siendo *vlimPQ(2)*, el inferior.

vlimPV: Ídem para nudos PV.

$$v\ limPQ = \begin{bmatrix} 1.05 \\ 0.85 \end{bmatrix} \quad v\ limPV = \begin{bmatrix} 1.02 \\ 0.98 \end{bmatrix}$$

8.1.2.4. Datos económico-financieros

Para el cálculo del VAN se tendrán en cuenta:

vu: Vida útil de la inversión (en años).

ta: Tasa de actualización (5% en el ejemplo).

pKWh: Precio del Kilovatio-hora en el mercado eléctrico (en €).

8.1.2.5. Número máximo de bancos, número máximo de escalones y clase de escalón

Se fija una serie de propiedades para las distribuciones de condensadores:

nc: El número máximo de nudos candidatos a realizar en ellos la compensación (igual al número máximo de bancos).

num_pasos: El número máximo de escalones o pasos de compensación, igual para todos los bancos.

KV_T: Para cada tensión nominal, se fija una clase de escalón entre las existentes en un fichero que se verá enseguida (condensadores.txt). Esta información se retiene en *KV_T*, matriz de dos columnas, que asocia para cada nivel de tensión en kV (primera columna) una clase de escalón (segunda columna), que viene dada por un valor escogido en una serie de números naturales. En el ejemplo de la figura 8.2, la matriz sólo presenta una fila:

$$KV_T = [23 \ 1]$$

8.1.2.6. Parámetros de funcionamiento del AG

Se agrupan bajo esta denominación los parámetros genéticos del AG:

num_ind: Tamaños de las poblaciones (en el ejemplo, 20 individuos).

num_gen: Número máximo de generaciones (10).

pcr, pmu: probabilidades de cruce y mutación (0.98 y 0.01).

nm, np: Tamaños de las subpoblaciones de los mejores y de los peores individuos de la última población generada (5 y 1).

elite: Indicador de elitismo (en el ejemplo toma valor 1, lo que significa que se seguirá un modelo elitista. En caso contrario, se fija un valor distinto de 1).

umbral_div: Umbral de diversidad genética para el que se pone fin a la evolución (15 %).

8.2. Función LEECONDENSADORES

Lee el fichero que contiene información sobre las potencias nominales

de los escalonamientos para cada clase de condensador y nivel de tensión. Incluye también los precios correspondientes.

```
*FICHERO CONDENSADORES.TXT
*
*** Coste fijo, potencia nominal(MVAr) y precio(euros) por paso, para
cada nivel de tensión(KV)
23 1145 0.3 1030 0.6 2060 0.9 3090 1.2 4120 1.5 5150
0
6 825 0.1 286 0.2 572 0.3 858 0.4 1144 0.5 1430
0
```

Figura 8.3. Ejemplo de fichero condensadores.txt.

La función LEECONDENSADORES presenta la siguiente sintaxis:

```
[DC]=leecondensadores(nombre_fichero)
```

8.2.1. Variable de entrada

nombre_fichero: 'condensadores.txt' o cualquier otro fichero con idéntica estructura.

8.2.2. Variable de salida

condensadores.txt es un fichero modificable también por el usuario, que contiene los valores disponibles a priori para los pasos de compensación. La información contenida en el fichero de la figura 8.3 se debe interpretar en la forma siguiente:

Nivel de tensión (kV)	Coste fijo	Potencia nominal y precio por escalón (MVAr, €) para cada clase .							
		Clase '1'		Clase '2'		Clase '5'		
		MVAr	€	MVAr	€	MVAr	€	
23	1145	0.3	- 1030	0.6	- 2060	1.5	- 5150	
6	825	0.1	- 286	0.2	- 572	0.5	-1430	

Para cada nivel de tensión, se tiene la posibilidad de seleccionar una clase de escalón entre las 5 existentes. Así, para un nudo candidato 'i'

a 23 kV, una vez elegida la clase '1' (en el fichero interfaz.txt) para todos los nudos a 23 kV, se tendrá un banco con x_i escalones clase '1', con una potencia nominal total de $x_i * 300 \text{ kVAr}$ y con un precio total igual a $1145 + x_i * 1030 \text{ €}$.

Los costes por escalón de la tabla se refieren a condensadores fijos. Para condensadores conmutables se considera un precio por escalón un 100% superior. Para un banco como el del ejemplo, esta vez con ajuste automático, el coste del dispositivo ascendería a $1145 + x_i * 2 * 1030 \text{ €}$.

La información almacenada en el fichero se vuelca en:

DC: Matriz cuyas columnas son: nivel de tensión(KV), coste fijo de instalación(€), potencia nominal (MVar), precio (€) por escalón-clase 1-, potencia nominal(MVar), precio(euros) por escalón-clase 2-, ...

Así, para el fichero representado en la figura 8.3:

$$DC = \begin{bmatrix} 23 & 1145 & 0.3 & 1030 & 0.6 & 2060 & \dots & 1.5 & 5150 \\ 6 & 825 & 0.1 & 286 & 0.2 & 572 & \dots & 0.5 & 1430 \end{bmatrix}$$

8.3. Función LEECASO

Cada vez que se llama esta función, se lee un fichero .raw. En la figura 8.4 se reproduce un fichero ejemplo: TS1S1.raw.

La sintaxis de la función LEECASO es la siguiente:

```
[base,n,tipo,nivelv,v,d,Pd,Qd,Pg,Qg,Qmin,Qmax,vconsig,TAD,Y,BrD,SHUNT]
=leecaso(nombre_fichero)
```

8.3.1. Variable de entrada

nombre_fichero: Nombre del fichero .raw, por ejemplo, 'TS1S1.raw'.

8.3.2. Variables de salida

La descripción de los casos base viene dada en forma de ficheros en

formato PTI Raw Data (versión 24), con extensión *.raw. Son tipo ASCII, por lo que se pueden leer y modificar con cualquier editor de textos (Notepad, Word, editor de Matlab). El formato es el estándar que emplean los programas de simulación de sistemas de potencia.

Estos ficheros presentan la siguiente estructura:

8.3.2.1. Datos de identificación del caso

El fichero presenta una cabecera con la descripción del caso. Entre los datos que figuran en ella, se lee la potencia base (MVA) para el sistema en por unidad y la almacena en la variable:

base: Potencia base (MVA). En TS1S1.raw, *base* = 10 MVA.

```

0      10.00          / marzo 10, 2002 13:22:41
Sistema Radial de 9 ramas.TS1S1
Escenario Escalado 0.6 TS1
  1 'FIN      ' 23.00 1      0.00      0.00      1      1 0.82918      -5.5945
  2 '2        ' 23.00 1      0.00      0.00      1      1 0.85055      -4.9977
  3 '3        ' 23.00 1      0.00      0.00      1      1 0.88105      -4.2001
  4 '4        ' 23.00 1      0.00      0.00      1      1 0.90077      -3.8633
  5 '5        ' 23.00 1      0.00      0.00      1      1 0.91173      -3.5072
  6 '6        ' 23.00 1      0.00      0.00      1      1 0.94472      -2.5617
  7 '7        ' 23.00 1      0.00      0.00      1      1 0.96086      -2.2838
  8 '8        ' 23.00 1      0.00      0.00      1      1 0.98625      -1.2609
  9 '9        ' 23.00 1      0.00      0.00      1      1 0.99247      -0.5163
 10 '10       ' 12.66 3      0.00      0.00      1      1 1.00000      0.0000
0
  1 1 1 1 1      0.984      0.120
  2 1 1 1 1      0.588      0.078
  3 1 1 1 1      0.690      0.360
  4 1 1 1 1      0.468      0.066
  5 1 1 1 1      0.966      0.360
  6 1 1 1 1      0.959      1.104
  7 1 1 1 1      1.074      0.268
  8 1 1 1 1      0.588      0.204
  9 1 1 1 1      1.104      0.276
0
 10 1      13.180      5.814 990.000 -990.000 1.00000      0 100.000 0.00000 1.00000
0.00000 0.00000 1.00000 1 100.0 100.000      0.000
0
  2      1 1      0.10100 0.05720 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
  3      2 1      0.09060 0.05130 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
  4      3 1      0.03890 0.02200 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
  5      4 1      0.01710 0.01490 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
  6      5 1      0.03750 0.03270 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
  7      6 1      0.01320 0.01150 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
  8      7 1      0.01410 0.02280 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
  9      8 1      0.00030 0.01140 0.00000 1000.0 1000.00      0.00,,,      0.00000      0.00000
0.00000 0.00000 1      0.0
 10      9 1      0.00230 0.00780 0.00000 1000.0 1000.00      0.00 1.00000      0.000
0.00000 0.00000 0.00000 0.00000 1      0.0
0
 10      9 1      10 1.10000 0.90000 1.00000 0.99000 0.00625 0 0 0.00000
0.00000
0
  1      0      0.000      0.100 '1      '
0
0
0
0
0
0
0
0
  1 '1      '
0
0

```

Figura 8.4. Ejemplo de fichero caso_base.raw: TS1S1.raw.

8.3.2.2. Datos de los nudos

Para cada nudo de la red se tiene una línea de texto con el siguiente formato:

I, NAME, BASKV, IDE, GL, BL, IA, ZONE, VM, VA

Se describen a continuación sólo aquellos campos que son leídos por la función LEECASO (tanto en este bloque como en los siguientes), indicando las correspondientes variables del programa.

n : Número total de nudos de la red. Se obtiene contando el número de filas existentes en el bloque ($n = 10$ en TS1S1).

Los índices de los siguientes vectores están asociados con los índices (I) de los nudos:

$nivelv$ (BASKV): Vector (nx1). Tensión de los nudos en kV.

$tipo$ (IDE): Vector (nx1) cuyos elementos toman los valores 1, 2 ó 3, según se trate de un nudo tipo PQ, PV, o slack, respectivamente.

v (VM), d (VA): Vectores (nx1) que contienen el módulo en p.u., $v(j)$, y el argumento en grados, $d(j)$, de la tensión de cada nudo j . Para el ejemplo:

$$\begin{array}{l}
 nivelv = \begin{bmatrix} 23 \\ 23 \\ 23 \\ 23 \\ 23 \\ 23 \\ 23 \\ 23 \\ 23 \\ 12.66 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 tipo = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 3 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 v = \begin{bmatrix} 0.8292 \\ 0.8506 \\ 0.8811 \\ 0.9001 \\ 0.9117 \\ 0.9447 \\ 0.9609 \\ 0.9863 \\ 0.9925 \\ 1.0000 \end{bmatrix}
 \end{array}
 \begin{array}{l}
 d = \begin{bmatrix} -5.59 \\ -5.00 \\ -4.20 \\ -3.86 \\ -3.51 \\ -2.56 \\ -2.28 \\ -1.26 \\ -0.52 \\ 0.00 \end{bmatrix}
 \end{array}$$

SHUNT: Matriz de pares nudo-admitancia compleja shunt conectada al nudo. No incluye las admitancias paralelo de las líneas y trafos. En TS1S1 es una matriz nula. Se obtiene a partir de los pares (*GL*, *BL*), que son las componentes real e imaginaria de la admitancia expresadas en MW y MVar, respectivamente.

8.3.2.3. Datos de los consumos

Cada línea de este bloque se corresponde con la información asociada a una carga. El formato es:

I, ID, STAT, IA, ZONE, PL, QL

Los índices de los siguientes vectores están asociados con los índices (I) de los nudos:

Pd (*PL*), *Qd* (*QL*): Vectores (n x 1) que recogen las potencias activas (MW) y reactivas (MVar) demandadas por los n nudos. Para TS1S1:

$$Pd = \begin{bmatrix} 0.984 \\ 0.588 \\ 0.690 \\ 0.468 \\ 0.966 \\ 0.959 \\ 1.074 \\ 0.588 \\ 1.104 \\ 0.000 \end{bmatrix} \quad Qd = \begin{bmatrix} 0.120 \\ 0.078 \\ 0.360 \\ 0.066 \\ 0.360 \\ 1.104 \\ 0.268 \\ 0.204 \\ 0.276 \\ 0.000 \end{bmatrix}$$

8.3.2.4. Datos de los generadores

Cada línea de este bloque se corresponde con la información asociada a un generador. El formato es:

I, ID, PG, QG, QT, QB, VS, IREG, MBASE, ZR, ZX, RT, XT, GTAP, STAT, RMPCT, PT, PB

Los índices de los siguientes vectores están asociados con los índices (I) de los nudos:

Pg (PG), Qg (QG): Vectores (nx1) que recogen las potencias activas (MW) y reactivas (MVA_r) generadas en los n nudos.

Qmin (QB), Qmax (QT): Vectores (nx1) Límites inferior y superior, respectivamente, de la potencia reactiva (MVA_r) que pueden generar.

Vconsig (VS): Vector (nx1). Representa la tensión de consigna o 'set-point'(p.u.). Cuando el nudo es de consumo, toma valor nulo.

En el ejemplo:

$Pg(10) = 13.180$, $Qg(10) = 5.814$, $Qmin(10) = -990$, $Qmax(10) = 990$,
 $Vconsig(10) = 1.0000$

8.3.2.5. Datos de las ramas

Para cada rama de la red se tiene una línea de texto con el siguiente formato:

I, J, CKT, R, X, B, RATEA, RATEB, RATEC, RATIO, ANGLE, GI, BI, GJ, BJ, ST, ?

La información relevante se guarda en:

BrD: Matriz que describe en cada fila las características de cada rama: nudo de salida (I), nudo de llegada (J), resistencia (R), reactancia (X) y

susceptancia total (B) en p.u., relación de transformación en módulo (RATIO) en p.u. y argumento (ANGLE) en grados si es trafo (sino, ambos valen cero), identificador de circuito (CKT), flujo de potencia máximo (RATEA) en MVA, admitancia paralelo del lado I y del lado J, admitancia serie IJ y admitancia serie JI.

Ybus: Matriz (nxn) de admitancias de nudos del caso base.

8.3.2.6. Datos de control automático de los transformadores

Cada línea de este bloque se corresponde con la información asociada a un transformador. El formato es:

I, J, CKT, ICONT, RMA, RMI, VMA, VMI, STEP, TABLE, CNTRL, ?, ?

Se construye la matriz:

TAD: Matriz de datos de control de los transformadores. Cada fila contiene la información correspondiente a cada uno de ellos, a saber: nudo de salida (I), nudo de llegada (J), identificador de circuito (CKT), nudo de tensión controlada (ICONT), máxima y mínima relación de transformación (RMA Y RMI), máxima y mínima tensión deseables en el nudo controlado (VMA Y VMI), paso (STEP), "flag" de control automático (CNTRL:1-sí, 0-no), relación de transformación en módulo (p.u.) y argumento (en grados), admitancia de cortocircuito.

8.3.2.7. Otros datos

El fichero contiene otros datos que no son relevantes para el problema.

8.3.2.8. Índices de los escenarios

En realidad, a las variables definidas en 8.3 hay que añadirles una dimensión relativa al índice del escenario, que coincide con el de la fila

que le corresponda en la matriz `casos_base`:

$$\text{casos_base} = \begin{bmatrix} TS1S2 \\ TS1S1 \\ TS1S0 \end{bmatrix}$$

Así por ejemplo, el vector que registra las potencias activas demandadas en el caso base `TS1S1` sería $Pd(:,2)$.

9. Módulo INICIO

En el módulo INICIO, se inician las variables internas del algoritmo y se definen las estructuras "individuo" empleadas por el algoritmo genético principal (AGP). Cada individuo es una posible solución del problema "max VAN", y un conjunto de ellos conforma una población. En cada iteración, se tendrá una población POBP constituida por los padres y NPOBP, correspondiente a los hijos.

Su diagrama de flujo es mostrado en la figura 9.1.

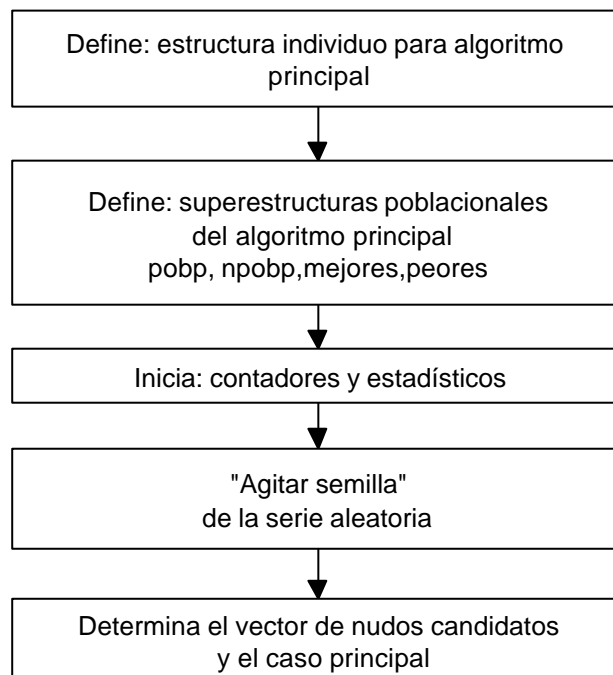


Figura 9.1. Módulo INICIO.

9.1. Caso principal y nudos candidatos

Por facilitar la exposición, se comienza por explicar el último bloque de la figura 9.1.

La determinación del caso principal y de los nudos candidatos se

realiza analizando los estados de los casos base, para lo cual se efectúa un reparto de carga sobre cada uno de ellos. Si los valores iniciales de la tensiones, potencias, etc., leídos de los ficheros .raw se ajustan al error máximo admisible, el Newton-Raphson debería converger sin realizar iteración alguna (ficheros con datos coherentes).

Para cada caso base, se calculan las pérdidas (vector Ploss0) y se establece según el criterio "pérdidas mayores" cuál va a ser el caso principal (cp).

Sobre el caso principal, se toma n_c nudos de consumo que presenten un nivel de tensión más bajo para constituir el vector de nudos candidatos, NdC (también se podría elegir los nudos de forma arbitraria). Supóngase, por ejemplo, que $n_c=3$ y los nudos en los que se permite ubicar bancos han resultado ser los nudos 3, 4 y 7. Entonces:

$$NdC = [3, 4, 7]$$

Determinado NdC, a partir de las variables nivelv, DC y KV_T (definidas en el módulo DATOS), se crean los vectores que recogen la potencia reactiva (Qip), precio fijo (pf) y precio (pp) por paso de los bancos en los nudos candidatos. Así, por ejemplo, si para $NdC = [3, 4, 7]$ se obtuviera:

$$Qip = [0.1, 0.6, 0.3], pf = [825, 1145, 1145], pp = [286, 2060, 1030]$$

se quiere decir que el banco situado en el nudo 3, está constituido por escalonamientos de 0.1 MVAR, con un coste de 286 € por escalonamiento para bancos fijos (286*2 para bancos variables) y un coste de instalación de 825 €, etc.

9.2. Definición de la estructura "individuo".

La estructura de un individuo para el AGP se compone de:

- Fenotipo: Distribuciones propiamente dichas, por ejemplo, 2 escalones de potencia en el nudo 3, 5 en el 4, 1 en el 7., en el

caso principal; 2 escalones en el 3, 3 en el 4, y 0 en el 7., para el caso secundario A; 1 escalón en el nudo 3, 2 en el 4, y 1 en el 7, para el caso secundario B; etc.

- Genotipo o cromosoma: Codificación binaria de la distribución principal y de las derivadas resultantes, proporcionadas por el AGB.
- Pérdidas: Valor de las pérdidas eléctricas en los distintos escenarios.
- Estado: Fasores de tensión de los nudos que describen los estados en los que queda el sistema eléctrico una vez ensayadas las distribuciones de condensadores en las situaciones de consumo consideradas.

La estructura individuo se implementa en Matlab de la forma siguiente:

1. Fenotipo: ind.x.

Una distribución viene representada por dos vectores. El vector fijo NdC, descrito en el apartado anterior, y otro variable:

$$X = \{x_1, x_2, \dots, x_j, \dots, x_{nc}\},$$

donde x_j es el número de escalones del banco ubicado en el nudo designado por NdC(j).

Entonces se construye la matriz ind.x con filas que son los vectores X de las distribuciones correspondientes a los distintos escenarios. La correspondencia se establece igualando el índice de la fila al índice del escenario.

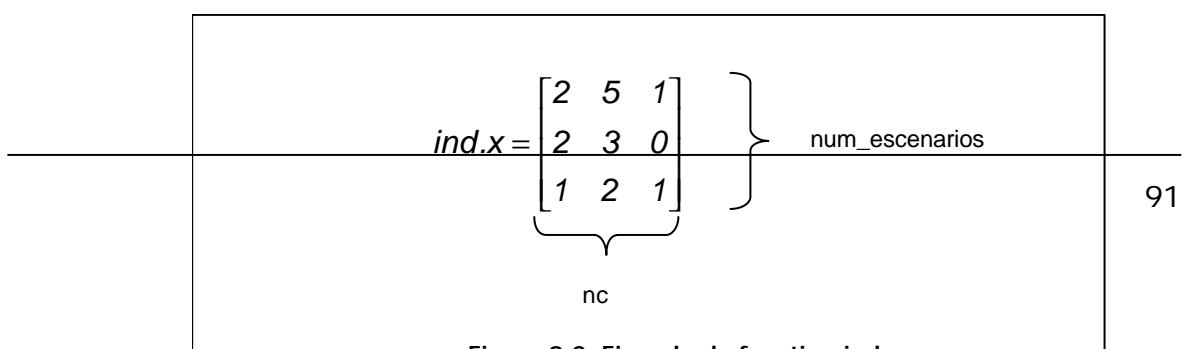


Figura 9.2. Ejemplo de fenotipo ind.v

2. Cromosoma: *ind.crom*.

Los cromosomas están constituidos por genes y cada gen es la codificación binaria del número de escalonamientos de un banco. La longitud del cromosoma (*lcr*) se determina por el producto de la longitud del gen (*lgen*) y el número de nudos candidatos (*nc*), siendo *lgen* función del número de pasos máximo (*num_pasos*) o escalones de potencias de los bancos de condensadores.

Así por ejemplo, si se consideran bancos de 7 escalones como máximo, para $X = [2 \ 5 \ 1]$, el primer gen viene dado por la cadena $[0 \ 1 \ 0]$, el segundo por $[1 \ 0 \ 1]$ y el tercero es $[0 \ 0 \ 1]$

ind.crom es una matriz cuyas filas son los cromosomas que codifican las distribuciones principal y secundarias (la correspondencia se establece igualando el índice de la fila al índice del escenario).

$$ind.crom = \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 1 \end{bmatrix} \left. \vphantom{\begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 1 \end{bmatrix}} \right\} \text{Num_escenarios}$$

$\underbrace{\hspace{10em}}_{lcr}$

Figura 9.3. Ejemplo de genotipo *ind.crom*.

3. Pérdidas: *ind.loss*.

ind.loss es un vector que registra las pérdidas (MW) para cada caso base modificado por las distribuciones correspondientes.

$$ind.loss = \begin{bmatrix} 1.400 \\ 0.530 \\ 0.150 \end{bmatrix} \left. \vphantom{\begin{bmatrix} 1.400 \\ 0.530 \\ 0.150 \end{bmatrix}} \right\} \text{ num_escenarios}$$

4. Bondad de la distribución. *Flujo de caja* *ind.van* de pérdidas *ind.loss*.

Es un escalar que resulta de calcular el VAN

$$ind.van = VAN(D, sum(N), ta, vu)$$

a partir de:

- La inversión (D) provocada por la distribución de condensadores

$$D = sum(ind.x(cp,:). *pp. *tipo_cond) + sum(sign(ind.x(cp,:)). *pf);$$

Donde *tipo_cond* es un vector que se obtiene de comparar las distribuciones principal y secundarias entre sí.

tipo_cond = 1, si el banco es de tipo fijo y *tipo_cond* = 2, si es variable.

- Los flujos de caja (N) anuales originados por las distribuciones en sus respectivos escenarios. Para el designado por el índice w:

$$N(w) = (Ploss0(w) - ind.loss(w)) * ha(w) * 1000 * pKWh$$

- La tasa de actualización (ta) y la vida útil (vu) de los condensadores, respectivamente, definidos en el módulo DATOS.

5. Estado: *ind.estado*

Matriz que contiene los módulos y desfases de la tensión en los nudos (véase figura 9.5).

$$ind.estado = \begin{bmatrix} v_1 & d_1 & \dots & \dots & v_1 & d_1 \\ v_2 & d_2 & \dots & \dots & v_2 & d_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ v_n & d_n & \dots & \dots & v_n & d_n \end{bmatrix} \left. \vphantom{\begin{bmatrix} v_1 & d_1 & \dots & \dots & v_1 & d_1 \\ v_2 & d_2 & \dots & \dots & v_2 & d_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ v_n & d_n & \dots & \dots & v_n & d_n \end{bmatrix}} \right\} n$$

Escenario 1
Escenario último

9.3. Poblaciones

Se construye dos poblaciones con la estructura individuo.

- Población padre: *pobp* (de "población principal").
- Población hija: *npobp* (de "nueva población principal").

El tamaño de las poblaciones permanece constante e igual a *num_ind*. *pobp(j)* y *npobp(j)* son los individuos que ocupan la posición *j* en ambas poblaciones.

9.4. Subpoblaciones

En cada generación se ordenan los individuos por su bondad y se crea dos subconjuntos poblacionales:

- El de los "mejores": los *nm* primeros.
- El de los "peores": los *np* últimos.

Resultarán útiles para aplicar distintas estrategias de selección y reproducción, como por ejemplo la denominada "elitismo" (se verá más adelante).

9.5. Estadísticos y contadores

- Contador de generaciones: *cg*.
- Suma de bondades poblacional: *suma*. Es un vector fila de *num_gen* (número total de generaciones) elementos. En cada generación se guarda el sumatorio de *pobp(j).van* extendido a toda la población. Su valor es empleado por el operador genético selección. (Véase método *rueda ruleta*, apartado 11.1).

- Mejor y peor individuo de cada generación: mejor_ind y peor_ind. Son dos vectores con tantos elementos como número de generaciones.
- Número de mutaciones: NMUT. Vector que acumula el número de mutaciones de bits efectuados en cada generación.
- Número de 'no convergencias': NOCONV. Acumula el número de veces que el algoritmo de Newton-Raphson no converge. La falta de convergencia se produce fundamentalmente por la elección de valores inadecuados para los escalones de compensación.

9.6. Agitar semilla de la serie aleatoria

Los procesos aleatorios se valen en realidad de una sucesión de números pseudoaleatorios. Para que el programa utilice una distinta cada vez, es preciso "agitar la semilla", es decir, proveer al sistema de un número arbitrario que sirva de inicio para un nuevo rosario de números "aleatorios". Se emplea para ello el reloj del ordenador en el que se ejecuta la aplicación.

10. Módulo POB_INICIAL

Se encarga de producir una población inicial para el AGP a partir de la generación aleatoria de los cromosomas de las distribuciones principales. Su diagrama de flujo se muestra en la figura 10.1.

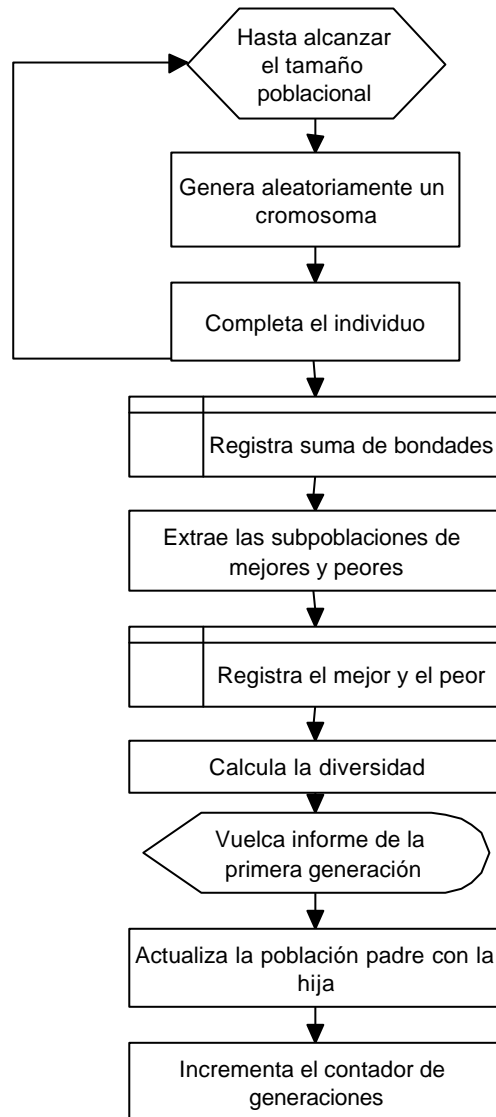


Figura 10.1. Módulo POB_INICIAL.

Tanto POB_INICIAL como EVOLUCION (se trata en el capítulo 11), utilizan las rutinas:

1. COMPLETA_IND.M
2. SUBPOBLACIONES.M
3. DIVERSIDAD.M
4. INFOR_GENERACIÓN.M

10.1. Módulo COMPLETA_IND

Su diagrama de flujo se muestra en la figura 10.2. Su tarea es la de completar la estructura del individuo a partir del conocimiento de su genotipo, generado éste ya sea en el módulo POB_INICIAL, o en el módulo EVOLUCION.

En una primera fase, el módulo comprueba que la distribución principal obtenida es una solución válida, previa llamada a la función REPARTO_CARGA. Si lo es, registra su fenotipo, las pérdidas y el estado del sistema originados por ella en el caso principal. También evalúa el flujo de caja anual asociado.

En una segunda etapa, se estudia qué distribuciones derivadas proporcionan los máximos flujos de caja en cada uno de los otros escenarios. Este análisis se hace, como se ha indicado ya, a través del Algoritmo Genético Básico (módulo AGB), sobre cada uno de los casos secundarios. Las soluciones obtenidas se incorporan a la estructura individuo y se determina finalmente el valor actual neto de la inversión asociada.

El AGB es una variante simplificada del algoritmo externo. Posee módulos análogos a los que utiliza el AGP, pero sus nombres aparecen precedidos por las letras AGB, esto es, AGB_INICIO, AGB_POB_INICIAL, AGB_EVOLUCION...

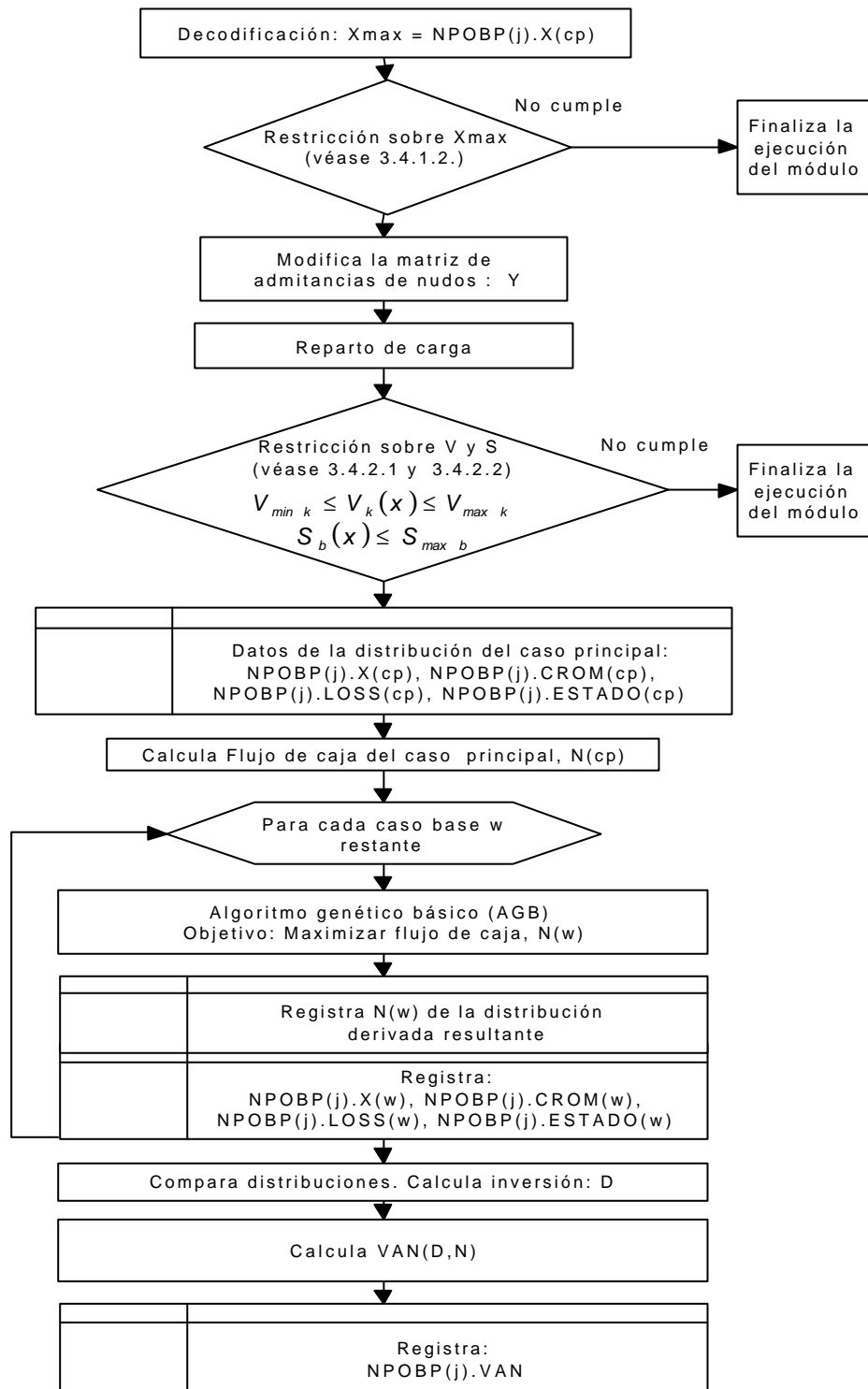


Figura 10.2. COMPLETE_IND.

Con objeto de acelerar el algoritmo, en el AGB se usará un “fenotipo reducido”: Se acorta la longitud de X (y por tanto del cromosoma) en caso de que en la distribución principal se presenten nudos sin escalones de potencia.

Finalmente se compara la distribución principal y las secundarias, para determinar los tipos de condensador resultante (tipo_cond) y calcular la inversión (D) y el VAN.

10.2. Módulo SUBPOBLACIONES

Construye las subpoblaciones descritas en 9.4. La utilidad de estas poblaciones se pondrá de manifiesto en el módulo EVOLUCIÓN.

10.3. Módulo DIVERSIDAD

Evalúa la diversidad genética poblacional, entendida ésta como la proporción de alelos distintos de los que constituyen el cromosoma del mejor individuo solución encontrado hasta la fecha.

Se utiliza el operador lógico XOR aplicado sucesivamente entre cada bit del mejor individuo y los bits que ocupen la misma posición en los restantes miembros de la población.

Se empleará como criterio de parada del programa el que la diversidad caiga por debajo de cierto valor (umbral_div), momento en el cual se puede considerar que la evolución está a expensas únicamente de las alteraciones que proporcione el operador mutación, que, por otra parte, tiene una probabilidad de actuación muy pequeña.

10.4. Módulo INFOR_GENERACIÓN

Muestra en pantalla la información más representativa sobre la última generación alumbrada:

- Fenotipo, VAN y pérdidas de todos los miembros de la población (véase figura 10.3).
- Mejor individuo.
- Diagrama de barras: índice individuo-van (figura 10.4).

ALGORITMO GENÉTICO PRINCIPAL										
ESCENARIO PRINCIPAL... TS1S0.raw										
Individuo n°:	Fenotipo:						VAN:	Pérdidas (en TS1S0.raw		
TS1S1.raw	TS1S2.raw):								
(1)	2	2	6	7	7	7	62049 euros	0.974 MW	0.228 MW	0.054 MW
(2)	6	4	2	7	1	7	60905 euros	1.002 MW	0.229 MW	0.054 MW
(3)	7	2	1	0	6	5	49348 euros	1.055 MW	0.234 MW	0.055 MW
(4)	7	6	0	7	4	3	51796 euros	1.024 MW	0.230 MW	0.055 MW
(5)	3	7	4	7	3	5	66157 euros	0.968 MW	0.228 MW	0.057 MW
(6)	5	5	4	2	5	6	63337 euros	0.995 MW	0.229 MW	0.055 MW
(7)	5	7	6	5	6	7	44892 euros	0.989 MW	0.229 MW	0.055 MW
(8)	1	4	6	2	7	7	63822 euros	0.987 MW	0.229 MW	0.054 MW
(9)	5	1	2	1	4	6	59786 euros	1.043 MW	0.233 MW	0.055 MW
(10)	0	4	7	7	1	2	80250 euros	0.983 MW	0.229 MW	0.055 MW
(11)	0	7	5	4	4	0	79010 euros	0.985 MW	0.230 MW	0.054 MW
(12)	0	7	1	7	3	2	79452 euros	1.001 MW	0.229 MW	0.054 MW
(13)	4	5	7	2	1	1	76730 euros	0.988 MW	0.230 MW	0.055 MW
(14)	1	5	5	7	5	7	64812 euros	0.968 MW	0.228 MW	0.055 MW
(15)	7	1	0	5	3	0	60179 euros	1.055 MW	0.234 MW	0.055 MW
(16)	7	0	7	4	1	3	62625 euros	1.016 MW	0.231 MW	0.056 MW
(17)	1	7	5	4	1	6	71882 euros	0.978 MW	0.229 MW	0.055 MW
(18)	7	3	4	0	3	7	54274 euros	1.032 MW	0.231 MW	0.055 MW
(19)	1	6	4	5	2	7	71286 euros	0.981 MW	0.228 MW	0.054 MW
(20)	7	0	7	4	1	2	67618 euros	1.017 MW	0.229 MW	0.055 MW
Mejor distribución encontrada en la 1ª generación:										
Individuo n°... 10										
Escenario:	Fenotipo:						Pérdidas:			
TS1S0.raw	0	4	7	7	1	2	0.983 MW			
TS1S1.raw	0	3	2	6	1	2	0.229 MW			
TS1S2.raw	0	1	1	1	1	0	0.055 MW			
VAN... 80250.2154euros.										
Diversidad... 46.6667 %										
K»										

Figura 10.3. Salida por pantalla realizada por el módulo INFOR_GENERACIÓN. Informe de la población obtenida en una iteración del AGP.

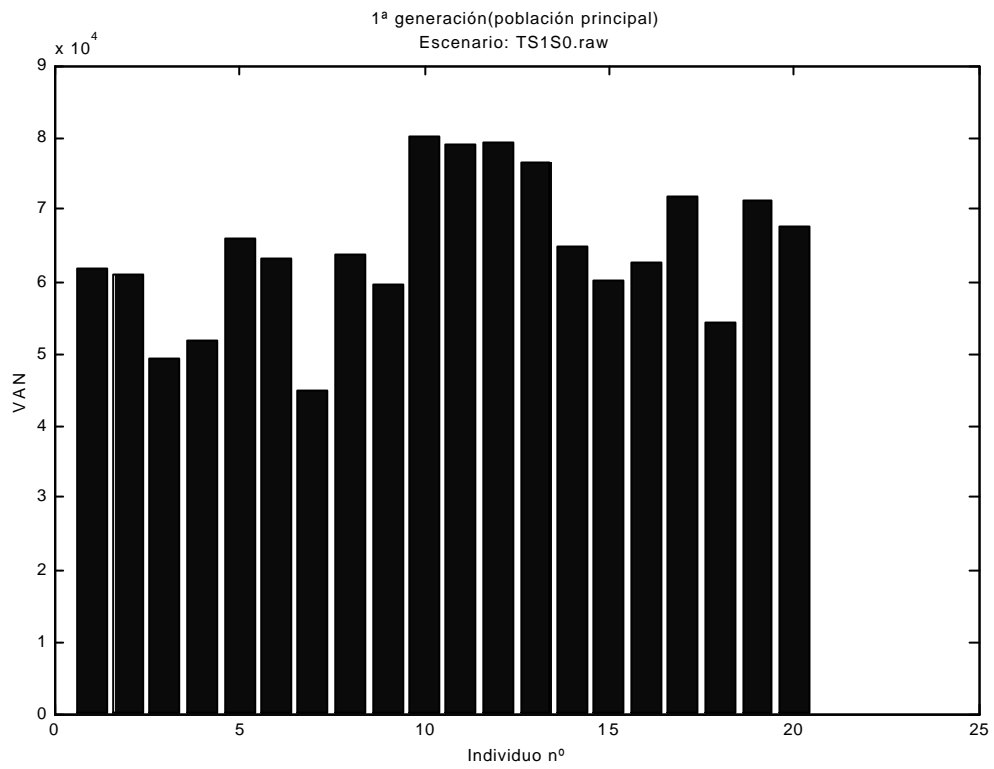


Figura 10.4. Salida por pantalla realizada por el módulo INFOR_GENERACIÓN. Diagrama de barras.

11. Módulo EVOLUCION

Presenta una estructura parecida al módulo POB_INICIAL (véase figura 11.1). La diferencia estriba en la forma en que se generan los individuos: Selección de una pareja, cruce de sus cromosomas y alguna mutación eventual (funciones SELEC, CRUCE y MUTA).

La evolución se produce asegurando la subsistencia de la mejor solución obtenida hasta el momento.

El bloque denominado 'elitismo' trabaja de la siguiente manera: si *elite* =1 (opción del usuario) se incluye directamente en la población hija (de la generación siguiente) los 'nm' mejores individuos de la última población producida en este módulo, con lo cual, se generan sólo num_ind - nm en la siguiente iteración.

11.1. Función SELEC

Se implementa la selección de "rueda ruleta" (véase figura 11.2) y la selección por "torneo". La segunda se usará únicamente cuando se detecte la existencia de valores de bondad negativos, lo que es previsible que ocurra en los primeros estadios evolutivos. Se utilizan técnicas de escalado. La función presenta la siguiente sintaxis:

```
y=selec(b, sb)
```

Variables de entrada:

b: Vector de bondades poblacional.

sb: Suma de las bondades.

Variables de salida:

y: Posición del individuo seleccionado en la población.

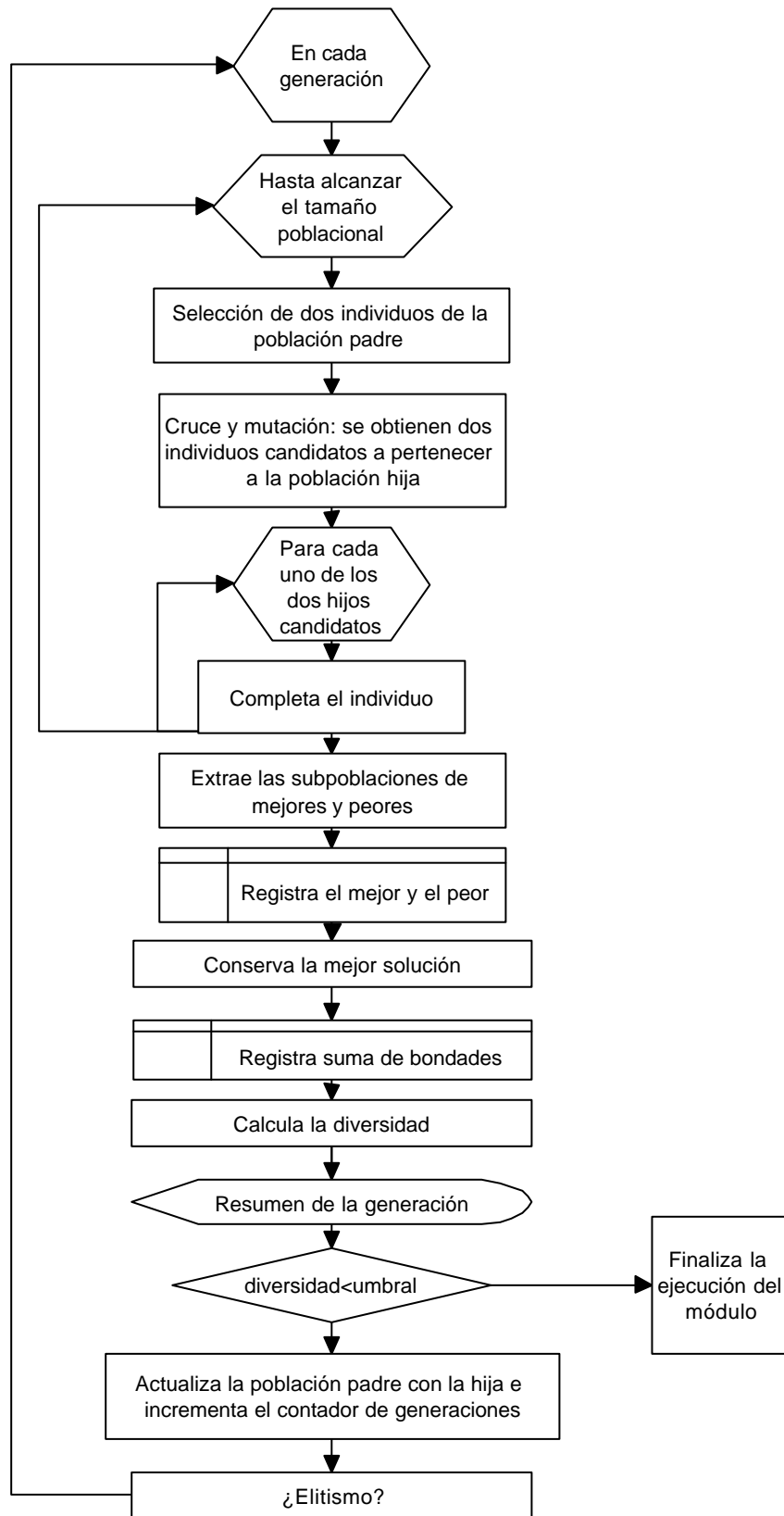
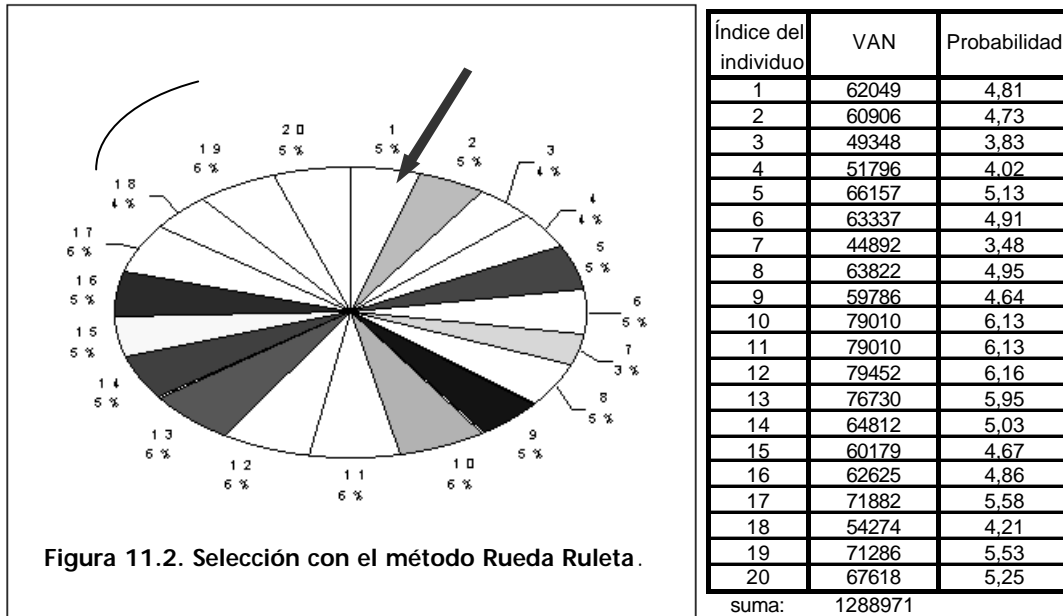


Figura 11.1. Módulo EVOLUCION.



La selección por el método "rueda ruleta" consiste en empezar generando un número aleatorio entre 0 y la suma de las bondades (sb): punto de parada (PP). A continuación se realiza la suma acumulada (SumAc) de las bondades desde el individuo '1' hasta el que hace que se cumpla: $\text{SumAc} > \text{PP}$, siendo entonces éste el que resulta seleccionado. De esta forma la probabilidad de cualquier individuo para ser padre, resultaría directamente proporcional a su bondad (figura 11.2). No obstante, con objeto de fomentar la diversidad poblacional y evitar una rápida convergencia del algoritmo, se emplea un escalado exponencial de la función objetivo: La bondad de los individuos se eleva a un exponente k . Un k mayor que uno agranda las diferencias entre las bondades, mientras que un k menor que uno, las reduce. En las primeras generaciones interesa un k menor que uno, y un k mayor que uno hacia el final. Se ensaya en el AG el siguiente escalado:

```
k = (num_gen+cg)/(1.5*num_gen);%Factor de escala para cg<(num_gen/2)
```

```
k = (num_gen+0.5*cg)/num_gen); %Factor de escala para cg>(num_gen/2)
```

La selección de torneo consiste en extraer aleatoriamente y sin reposición un número individuos equivalente al 10% de la población, seleccionando entre ellos aquél que tenga la mayor puntuación. Este porcentaje puede modificarse directamente en el código (línea 43 del fichero 'selec.m'):

```
if uoc(0.10)
```

11.2. Función CRUCE

La operación de cruce se efectúa cortando en un punto del cromosoma elegido al azar e intercambiando los alelos (bits) resultantes. Incorpora en su interior llamadas a la función MUTA.

```
[a,b]=cruce(p,q,pc,pm,lgen)
```

Variables de entrada:

p, q: Cromosomas seleccionados.

pc, pm: Probabilidades de cruce y mutación.

lgen: Longitud de un gen del cromosoma.

Variables de salida:

a, b: Cromosomas resultantes del cruce y la mutación de *p* y *q*.

11.3. Función MUTA

Implementa la función negación lógica sobre un alelo (bit) del cromosoma. Se realiza con una probabilidad fijada por el usuario (*pmu*).

```
y = muta(bit,pmu)
```

Variables de entrada:

bit: alelo del cromosoma.

pmu: probabilidad de mutación.

Variable de salida:

y: bit mutado con una probabilidad dada por *pmu*.

12. Módulo RESULTADOS

Se encarga de reflejar en el fichero 'solucion.txt' (véase Apéndice III) las mejores distribuciones de condensadores encontradas por el programa. Dado que éste se puede arrancar varias veces con distintos datos de entrada, se ha creído oportuno copiar en él los datos contenidos en el fichero 'interfaz.txt' empleado.

También muestra gráficamente en pantalla la evolución poblacional. Este gráfico se guarda en un fichero de formato binario, 'graf_evo.mat', y se puede sacar por pantalla con 'graf_evo.m'.

```

*****
*   Solución 1   *
*****

*** Distribución de condensadores
Escenarios:      TS1S0.raw      TS1S1.raw      TS1S2.raw
(Nudo)          (MVar nominales)
 1          0.600          0.300          0.300
 2          0.000          0.000          0.000
 3          1.200          0.300          0.300
 4          0.300          0.300          0.300
 5          1.200          0.300          0.600
 6          0.600          0.600          0.000

Escenarios:      TS1S0.raw      TS1S1.raw      TS1S2.raw
*** Pérdidas:    1.002          0.213          0.040
*** Valor Actual Neto de la Solución 1 : 124754

Estado:
NUDO      v(p.u.)      d(°)      v(p.u.)      d(°)      v(p.u.)      d(°)
 1      0.790      -9.17      0.925      -4.14      0.974      -2.59
 2      0.812      -8.11      0.934      -3.67      0.977      -2.28
 3      0.845      -6.86      0.950      -3.11      0.984      -1.93
 4      0.865      -6.08      0.959      -2.80      0.988      -1.71
 5      0.875      -5.47      0.964      -2.52      0.989      -1.54
 6      0.905      -3.69      0.978      -1.75      0.993      -0.97
 7      0.920      -3.09      0.985      -1.47      0.996      -0.78
 8      0.942      -1.58      0.996      -0.75      0.999      -0.38
 9      0.946      -0.67      0.997      -0.32      0.999      -0.17
10      1.000      0.00      1.000      0.00      1.000      0.00

*** ESTADÍSTICAS:

Número total de mutaciones: 2126
Número total de 'no convergencias': 0
Número de generaciones: 10, de un máximo de 10

```

Figura 12.1 Fragmento del fichero solucion.txt.

13.Módulo AGB

El Algoritmo Genético Básico (AGB) ajusta los escalones de potencia de los bancos que componen cada distribución principal (Xmax) con el objetivo de minimizar las pérdidas en los casos secundarios.

El AGB trabaja con distribuciones derivadas (secundarias) de una principal, suponiendo en principio que todos los bancos son regulables.

Está formado por un módulo inicial, AGB_INICIO, y sendos módulos AGB_POB_INICIAL y AGB_EVOLUCION, con las mismas cometidos que sus homólogos en el AGP. Estos últimos llaman a su vez a AGB_COMPLETA_IND y a AGB_INFORGENERACION.

De la misma forma que el AGP, presenta un informe final por pantalla (módulo AGB_RESULTADOS).

En AGB_INICIO se define una nueva estructura "individuo", sobre la que se construyen las poblaciones secundarias POBS y NPOBS (padre e hija, respectivamente). Los campos de la estructura son:

- Genotipo: Codificación binaria de la distribución secundaria. El cromosoma se acorta si la distribución principal presenta genes constituidos por alelos que son todos 0.
- Fenotipo: Distribución propiamente dicha, por ejemplo, 2 escalones en el nudo 3, 1 en el 4, etc.
- Pérdidas: Valor de las pérdidas eléctricas en el escenario secundario.
- Estado: Fasores de tensión de los nudos que describen el estado en el que queda el sistema eléctrico tras la instalación de los

condensadores que integran la distribución.

- Flujo de caja: Valor calculado a partir de los decrementos de las pérdidas eléctricas con respecto al caso base (sin condensadores) correspondiente.

El diagrama de flujo de AGB_COMPLETA_IND se presenta en la figura 13.1.

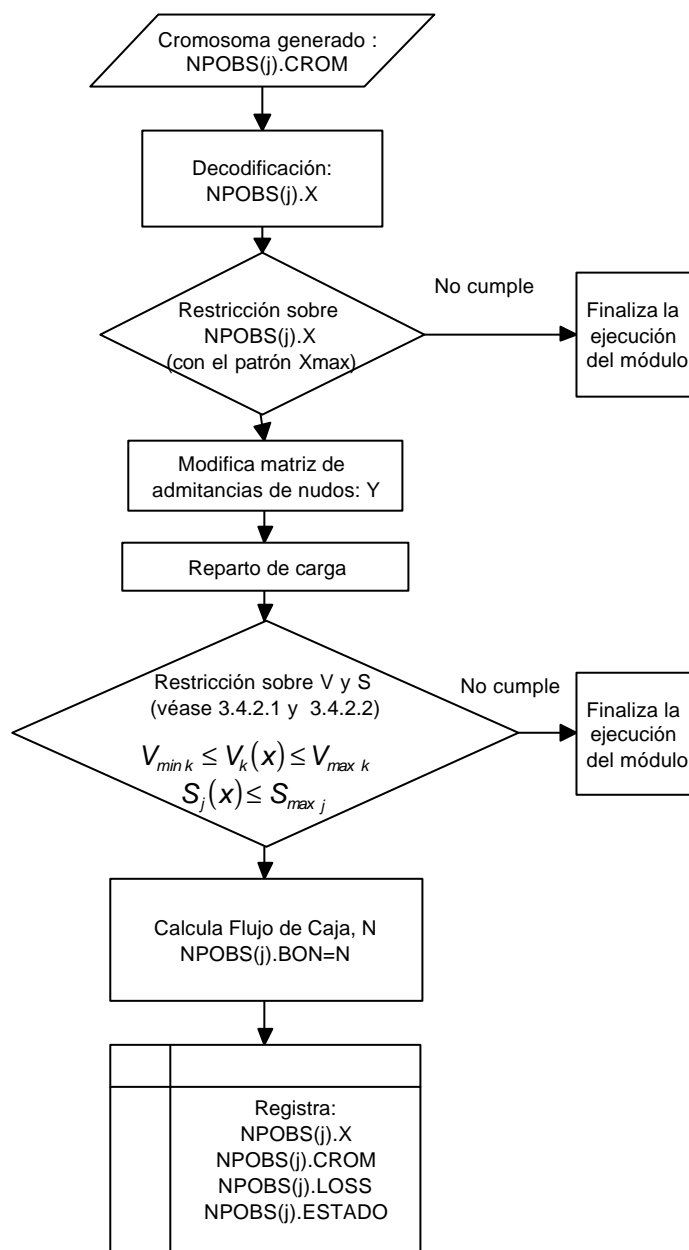


Figura 13.1. AGB_COMPLETA_IND.

