

# PREÁMBULO

El presente documento constituye el Proyecto Fin de Carrera que lleva por título "Algoritmo genético para la optimización de la inversión en bancos de condensadores en base a la evolución de la demanda en sistemas de potencia".

En este proyecto se investiga un método alternativo que permita decidir la localización, la potencia y el tipo de los bancos de condensadores que deben ser instalados en los nudos de una red eléctrica cualquiera para reducir, con una mínima inversión, los costes de explotación a lo largo de un horizonte temporal.

Se ensaya un algoritmo genético en la resolución del problema.

La exposición se estructura en cinco secciones:

**Introducción.** Se discute las implicaciones asociadas a la existencia de cargas reactivas en un sistema de potencia. Se describe en términos generales la técnica que se va a seguir en la resolución del problema (fundamentos de los algoritmos genéticos).

**Memoria descriptiva.** Se establece los términos en los queda planteado el problema y las líneas maestras del algoritmo.

**Memoria justificativa.** Se implementa el algoritmo genético; se trata con detalle los módulos principales del programa.

**Protocolo de pruebas.** Se analiza el comportamiento del programa sobre una red radial de diez nudos.

**Resultados y conclusiones.** Se ensaya nuevamente sobre una segunda red de distribución, esta vez de sesenta y nueve nudos; se presenta los resultados obtenidos y se extrae las conclusiones oportunas. Aún cuando la experimentación se realiza con redes radiales, el método

propuesto se diseña para ser utilizado en la planificación de cualquier tipo de sistema de potencia.

**Apéndices.** Se muestra íntegramente el código de programación en lenguaje Matlab, versión 5.1, así como los distintos tipos de ficheros de entrada y salida empleados.

**Manual de usuario.** Se proporciona la información necesaria para ejecutar correctamente el programa.

# INTRODUCCIÓN

## 1. Energía reactiva y pérdidas en las redes eléctricas

El desfase entre la tensión y la intensidad originado por la existencia de elementos no puramente resistivos en un sistema eléctrico, provoca la aparición de la componente reactiva del fasor Potencia Compleja,  $S$ , que se denomina Potencia Reactiva,  $Q$ .

Se trata de una potencia que es absorbida y cedida alternativamente por receptores (cargas inductivas en su mayoría) que la emplean para crear los campos magnéticos que necesitan para su funcionamiento, pero que no genera un trabajo neto en un ciclo. La propia naturaleza inductiva de las líneas hace también que parte de la potencia que fluye por las mismas sea de tipo reactivo.

Este trasiego de energía "adicional" tiene efectos negativos tanto en el consumo energético, como en la seguridad, fiabilidad y estabilidad del sistema eléctrico.

Los inconvenientes del flujo de potencia reactiva en la red de transporte eléctrico se pueden resumir principalmente en los siguientes:

1. Disminución de la capacidad estática de transmisión de las líneas, es decir, la potencia activa máxima que son capaces de transportar.
2. Aumento de la caída de tensión en las líneas.
3. Disminución de los márgenes de estabilidad dinámica.
4. Mayor riesgo de inestabilidad de las tensiones.

5. Aumentan las pérdidas al ser mayores las intensidades.

De todo esto se deduce la importancia de la compensación de la potencia reactiva. Ésta se puede efectuar en tres niveles:

**Nivel 1.-** En los centros de consumo, la compensación de reactiva permite rebajar el recibo de la compañía suministradora (recargo por reactiva para  $\cos\phi < 0.9$ ), aumentar la energía disponible en el secundario del transformador, reducir la sección de los conductores, así como las pérdidas por calentamiento en los mismos y en los transformadores [1].

**Nivel 2.-** En los nudos de las redes de distribución y reparto. De esta forma se disminuye la reactiva consumida desde la red de transporte, mejorándose el factor de potencia global.

**Nivel 3.-** En la propia red de transporte. Con ello se persigue corregir los inconvenientes mencionados arriba asociados al transporte de la potencia reactiva demandada, bien por las propias líneas, bien por elementos inductivos conectados en sus extremos, suministrándola directamente en algunos de los nudos de la red.

En el presente estudio se hará especial hincapié en las implicaciones sobre las pérdidas eléctricas y en la forma de reducirlas compensando la potencia reactiva en lo que se ha denotado como niveles 2 y 3. El método de resolución que se propone conducirá a dar respuesta a lo siguiente: ¿En qué nudos de la red es interesante ubicar los elementos compensadores de reactiva? ¿De qué tipo, fijo o variable? ¿Qué potencias nominales deben tener? Las soluciones que finalmente se consideren procederán del análisis de la inversión económica, asociada a la instalación de estos equipos, en comparación con los beneficios que reporte sobre el comportamiento del sistema eléctrico, primándose la disminución de las pérdidas y aceptando sólo estados posibles del sistema.

## 2. Introducción a los Algoritmos Genéticos

### 2.1. Qué se entiende por Algoritmo Genético

Los algoritmos genéticos (AAGG) son algoritmos de optimización: Tratan de resolver problemas del tipo "Dada una función  $F(x_1, \dots, x_n)$ , hallar las  $(x_1, \dots, x_n)$  tales que minimizan o maximizan  $F$ ".

Para explicar la técnica que usan para lograr su propósito, se toma como punto de partida la definición de D. Golberg [2]:

"Los Algoritmos Genéticos (AAGG) son algoritmos de búsqueda basados en los mecanismos de la selección natural y de la genética. Partiendo de estructuras tipo cadena, combinan la supervivencia de la más apta, con el intercambio estructurado, aunque aleatorizado, de información entre ellas, constituyendo así una búsqueda con algo en común a la forma en la que el hombre experimenta y progresa".

Los AAGG tratan de imitar a la Naturaleza en tanto que opera en busca del óptimo, produciendo como resultado especies perfectamente adaptadas a su entorno. Para entender cómo lo hace, se revisan brevemente las dos ideas clave: selección natural y genética [3].

- Charles Darwin ("El origen de las especies") descubrió el mecanismo que explicaba el fenómeno y que llamó **selección natural**: Sólo los individuos mejor dotados logran sobrevivir y reproducirse, heredando los hijos las buenas cualidades que permiten que la especie se perpetúe generación tras generación.

El proceso de transformación que sufren las especies está de acuerdo con esta teoría; en un medio ambiente cambiante, ciertas características individuales empezarían a cobrar mayor relevancia, en detrimento de otras, en el sentido de que aquellos ejemplares que las presentaran en mayor grado contarían con ventaja en la

lucha por alcanzar la edad adulta y reproducirse. De esta forma, tales cualidades individuales tenderían poco a poco a extenderse a la totalidad de la población. Tras siglos de evolución, los descendientes llegarían a diferenciarse tanto de sus primeros padres que darían lugar a una nueva especie.

- La Genética, posteriormente, explicó el por qué de la diversidad y cómo se produce la herencia: Toda la información acerca de la constitución física de un organismo vivo se encuentra codificada a nivel celular en una molécula denominada ADN. O dicho de otro modo, el ADN, que se encuentra formando parte de los cromosomas (el genotipo), controla la producción de enzimas que serán los responsables de la formación de los caracteres de cada individuo (el fenotipo). Así, se observaron dos hechos importantes:
  - Los cromosomas de dos individuos de la misma especie no son idénticos (diversidad).
  - Los cromosomas de las células sexuales de los padres son los encargados de transmitir las características propias de la especie, al mismo tiempo que sus manifestaciones peculiares (herencia).

¿Cómo se traduce todo esto en un algoritmo genético?

Definido el problema como:

$$\min \text{ ó } \max y = F(x_1, \dots, x_n)$$

sujeto a una serie de restricciones

1. Se identifica 'y' como la cualidad, o conjunto de ellas, que se pretende mejorar; las variables del problema,  $(x_1, \dots, x_n)$ , como el fenotipo de cada individuo; y 'F', como la función objetivo, de adaptación o bondad, que proporciona el grado de excelencia de cada individuo como solución del problema.

2. Se codifica el fenotipo en lo que Golberg llama estructura cadena, esto es, un cromosoma artificial (genotipo).
3. Se parte de un conjunto de posibles soluciones (población de individuos), que empiezan a competir para ver cuál constituye la mejor solución (aunque no necesariamente la mejor de todas las posibles). Los pares individuo-función objetivo  $F$  (ambiente), ejercerán una presión selectiva sobre la población, de forma que los que dan un mejor valor de  $F$  (los más adaptados) sobrevivan.
4. Estos últimos se emparejan, intercambiando parte de su información y dejando su legado genético a las futuras generaciones. La mejora continua debe estar sustentada por factores que fomenten la diversidad.

El azar juega un papel importante: La selección y el emparejamiento se desarrollan con un cierto grado de aleatoriedad. No obstante, los AAGG no son simples búsquedas azarosas (Goldberg habla de proceso "aleatorizado"). En lugar de ello, "explotan eficientemente la información del pasado para especular sobre nuevos puntos hacia los que dirigir la búsqueda, esperando mejorar su actuación" [2].

Para terminar de comentar la definición de Goldberg, este proceder evoca al modo en que el hombre realiza sus descubrimientos. En muchas ocasiones, el éxito viene precedido por la realización de múltiples ensayos, cuyos resultados son combinados de forma intuitiva para producir finalmente un nuevo hallazgo.

En este primer apartado se ha introducido los principales conceptos que se manejarán de aquí hasta el final. Queda por tratar varios temas; por ejemplo: ¿Cómo se va almacenar la información en el cromosoma? ¿Cómo se recrea la selección natural en un sistema artificial? ¿De qué forma se realiza la reproducción? ¿Cómo influye el azar en estos procesos? ¿Qué hacer para fomentar la diversidad imprescindible para conducir al óptimo? ¿Cuándo parar el algoritmo?

## **2.2. Características de los AAGG. Comparación con otros métodos de optimización**

Desde que se publicó la primera monografía al respecto [4], los AAGG han suscitado un gran interés en la comunidad científica, manifestado en la multitud de trabajos que prueban la validez de esta técnica como método de búsqueda y optimización.

Los AAGG presentan como virtudes el ser sencillos de implementar, el no estar comprometidos por restricciones sobre el espacio relacionadas con la continuidad y derivabilidad de la función objetivo, y, en muchos casos, disponer de un razonable grado de eficacia y eficiencia.

Existen otros métodos de optimización que se pueden clasificar en las siguientes categorías [2]:

- Basados en el cálculo.
- Numerales.
- Aleatorios.

Los basados en el cálculo han sido estudiados con profusión. Se subdividen en dos principalmente: Indirectos y directos. Los indirectos buscan el extremo local por resolución de un conjunto usualmente no lineal de ecuaciones que resultan de igualar el gradiente de la función objetivo a cero. Esta es la generalización multidimensional del problema del máximo o del mínimo del cálculo elemental: Dada una función continua y derivable, encontrar los posibles picos comienza por restringir la búsqueda a aquellos puntos donde la pendiente sea cero. Los métodos directos buscan el óptimo local saltando y moviéndose en una dirección relacionada con el gradiente local, "subiendo la montaña": Encontrar el mejor punto local recorriendo la función en la dirección permitida.

La principal desventaja de ambos métodos es que “miran” localmente; la búsqueda del óptimo se concentra en una región del espacio. Esto conduce a que no se encuentre en muchos casos el óptimo absoluto. Además, es imprescindible que la función sea derivable.

Los numerales consisten en buscar el óptimo recorriendo punto a punto todo el espacio. Es un método que no resulta eficiente en cuanto el número de puntos sea demasiado grande.

Los algoritmos de búsqueda aleatoria recorren la región de posibles soluciones al azar y guardan la mejor. Presentan el mismo inconveniente que los numerales.

Los AAGG emplean el azar para **guiar** la exploración del espacio, pero no son simples búsquedas aleatorias. En la Naturaleza, a la que tratan de imitar los AAGG, se producen fenómenos de forma aparentemente casual a pesar de contar con sus propias leyes.

Comparando los AAGG con otros métodos diseñados “ad hoc” para una determinada aplicación [5]:

- Las técnicas específicas suelen ser más rápidas.
- Los AAGG abordan el problema como si se tratara de una caja negra; básicamente les basta con conocer la expresión de la función que se desea optimizar. Esto no ocurre en los otros, donde un análisis más profundo se hace imprescindible para su diseño.
- El enfoque genético es de aplicabilidad general a cualquier tipo de problemas.
- Las etapas intermedias no son soluciones del problema en el método específico, pero sí lo son en un algoritmo genético: Se puede parar la evolución en cualquier momento y reanudarla a continuación si no se está satisfecho con las soluciones

obtenidas hasta ese instante.

- Los AAGG proporcionan una solución casi-óptima, que, muy probablemente, suele ser distinta cada vez. Se pretende, no obstante, que esté siempre cerca del óptimo (ajustando adecuadamente los parámetros de funcionamiento del AG).

En definitiva, los rasgos distintivos de los AAGG respecto a los otros métodos pueden resumirse en los siguientes:

1. Trabajan con un conjunto de valores codificados.
2. Buscan a partir de una población soluciones posibles.
3. Emplean directamente las funciones a optimizar, no sus derivadas ni ninguna otra información auxiliar.
4. Usan reglas de transición probabilística, no leyes deterministas.

### **2.3. Anatomía de un Algoritmo Genético Simple (AGS)**

Se continua la exposición planteando el diseño de un AG para resolver un problema de solución trivial, por lo que será evidente constatar si el algoritmo alcanza el óptimo. Así, por ejemplo, supóngase que se desea: *"Encontrar el entero  $x$  que proporciona el máximo de una función  $F(x)$  en un intervalo  $[a, b]$ ".*

Dada la generalidad de la que hacen gala los AAGG, no será complicado adaptarlo para resolver cualquier otro problema de optimización, en particular el que es objeto del presente proyecto.

*Se puede decir que el algoritmo genético trabaja con una materia prima que se procesa con la ayuda de unas herramientas.*

- **La 'materia prima': Población de individuos.**

Son un conjunto de candidatos a ser solución del problema. Se comienza por generar aleatoriamente una cantidad determinada de estos ejemplares. La única restricción que se les impone es que sean números enteros pertenecientes al intervalo  $[a, b]$ .

A cada uno de estos individuos se le asocia una estructura constituida por:

- A. El genotipo: Expresión del número entero en base 2. Se trata pues de una cadena binaria (de unos y ceros) o cromosoma que codifica el número entero. Continuando con el símil biológico, cada uno de los bits que la constituyen serían los alelos del cromosoma.
- B. El fenotipo: Apariencia externa del individuo, es decir, el propio número entero en base 10.
- C. Grado de adaptación o bondad: Evaluación de la función  $F(x)$  sobre cada individuo.

En definitiva, la población inicial, así como las sucesivas que se obtengan a partir de ella, son un conjunto finito y de tamaño constante de estructuras individuo, que se denota por **pob** (de población), siendo **pob(j)** el individuo  $j$ . Cada individuo  $j$  tiene un genotipo, **pob(j).crom**, un fenotipo, **pob(j).x**, y una bondad, **pob(j).bon**.

- **Las 'herramientas': Operadores genéticos**

El proceso evolutivo comienza con la selección de los individuos que serán los progenitores de la siguiente generación. La operación de selección consiste en extraer con reposición una pareja cada vez, de tal suerte que aquellos ejemplares que presenten una mayor bondad tendrán una probabilidad mayor de

ser seleccionados.

Las parejas seleccionadas se cruzan. La operación de **cruce** se efectúa con una cierta probabilidad intercambiando parte de los cromosomas en un punto arbitrario elegido al azar. De no producirse el cruce, ambos individuos pasan inalterados a la siguiente generación.

Se decide, adicionalmente, conservar al mejor espécimen hasta la fecha en las sucesivas generaciones. No hay que perder de vista que todos los esfuerzos deben ir encaminados hacia la mejora de la población en busca del óptimo.

Se realizará tantas veces como sea necesario para obtener un conjunto de descendientes válidos con igual tamaño al de la población inicial.

Para prevenir el eventual estancamiento de la población debida a la uniformidad genética que pueda obtenerse a lo largo del proceso evolutivo (sería una especie de consanguinidad por endogamia: el tamaño de la población es reducido), se introduce diversidad con un nuevo operador llamado **mutación**. Éste consiste en producir con una probabilidad baja el cambio de un 1 por un 0, o viceversa, en los alelos de la cadena cromosómica de cada nuevo individuo generado.

El AGS aparece implementado en lenguaje MATLAB, versión 5.1, en el Apéndice I. En Internet existen varios sitios web donde probar aplicaciones que resuelven éste y otros problemas de optimización y búsqueda. En concreto, en [6] se muestra un algoritmo genético, programado en Java por M. Ramsey (Universidad de Arizona), que trata también de hallar el máximo global de una función con muchos máximos locales.

Se ha probado el AGS con la función  $F(x)$ , definida en el archivo 'FUNC.M' sobre el intervalo  $[20,60]$ , con un máximo absoluto en 44

(figura 2.1). Dada la gran cantidad de máximos locales que presenta y lo próximo que se encuentran, encontrar el máximo se convierte en un problema complicado para cualquier procedimiento de optimización, incluido para el AGS (si no se toman las debidas precauciones).

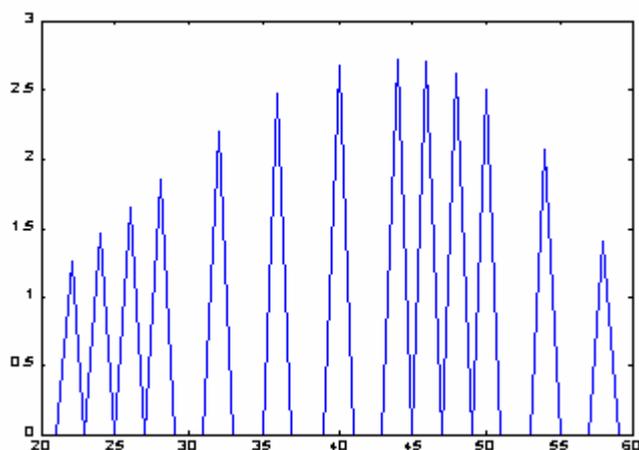


Figura 2.1. Función objetivo ensayada en el AGS,  $F(x)$ .

Tomando un tamaño poblacional de 8, con una probabilidad de cruce igual a 0.98, sin mutaciones y tras 15 generaciones, una primera ejecución del AGS dio como resultado:

» `ags(20,60,8,15,0.98,0.0)`

```

1.... 24 48 55 21 36 43 51 28
2.... 24 48 55 21 36 43 51 28
3.... 48 48 28 28 36 48 24 48
4.... 24 28 48 48 36 28 36 48
5.... 28 28 48 48 36 36 36 48
6.... 36 36 28 36 32 44 36 48
7.... 36 36 36 44 48 36 48 36
8.... 44 44 48 48 44 48 48 36
9.... 36 48 36 44 36 48 48 44
10... 36 44 36 48 44 48 44 44
11... 44 48 44 48 44 44 44 36
12... 44 44 48 36 44 44 44 44
13... 44 36 36 44 44 48 44 48
14... 44 44 48 36 44 36 44 44
15... 48 44 36 44 44 44 48 44

```

El entero perteneciente al intervalo (20,60) que ha sido hallado como el que da el mayor valor de la función implementada en `func.m`, ha resultado ser 44, tras 15 generaciones.

Número de mutaciones producidas: 0

En la figura 2.2 se representa la evolución.

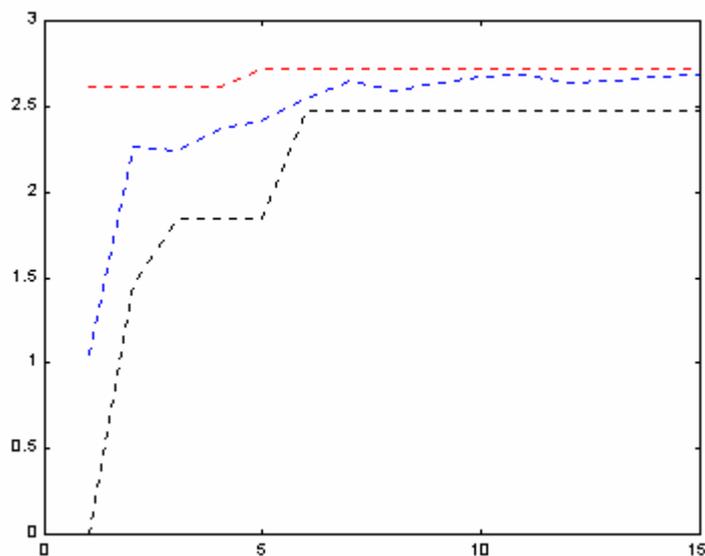


Figura 2.2. Evolución del valor de  $F(x)$ : Mejor individuo, valor medio y peor individuo

Se ha tomado la probabilidad de mutaciones igual a cero para demostrar que es posible hallar el óptimo simplemente por cruce de cromosomas.

Sucesivas actuaciones del algoritmo (con los mismos parámetros) no siempre obtuvieron  $x_{max}=44$ , pero con frecuencia encontraron un valor bueno. Sin embargo, para un problema tan sencillo como éste, el óptimo se alcanza prácticamente siempre si se permite las mutaciones, se aumenta el tamaño poblacional o el número de generaciones

#### 2.4. Decisiones para implementar un AG

En el apartado precedente se ha ilustrado la implementación de un AG con un ejemplo, que constituye en sí mismo un esquema básico para la implementación de otros AAGG.

Se examinan a continuación qué factores intervienen en el diseño de un AG [7].

### 2.4.1. Criterio de codificación

Se ha señalado ya la necesidad de codificar las variables del problema en lo que se ha denominado cromosoma. En el ejemplo 2.3, era algo tan simple como la representación binaria del entero perteneciente al intervalo  $[a, b]$ . Pero dado que el dominio está constituido por  $a-b+1$  individuos, el número de bits o posiciones necesarias,  $n$ , sería el resultante de redondear al alza el obtenido a partir de  $2^n = b-a+1$ . Con ello, la codificación adoptada inicialmente sufre un desplazamiento, dejando de ser la expresión directa en base 2.

Se observa entonces que:

- Se debe procurar que todos los bits de un individuo sean significativos. En pro de la eficiencia, la longitud del cromosoma ( $l_{cr}$ ) será lo más pequeña posible: El tiempo que consume un AG es exponencial con respecto a la longitud del cromosoma.
- Con la codificación se puede forzar intrínsecamente el cumplimiento de la restricciones del problema, en este caso que  $x$  pertenezca a  $[a, b]$ .

En cuanto a la longitud del cromosoma, ésta dependerá del número de variables y de la precisión con la que se trabaje. Se emplea el término "gen" para referirse al fragmento del cromosoma que codifica cada uno de ellas. En el AGS de 2.3, el cromosoma era mono-génico.

Hay muchas otras codificaciones posibles, empleando incluso alfabetos de diferente cardinalidad. No obstante, uno de los resultados fundamentales en la teoría de algoritmos genéticos, el teorema de los esquemas [4], afirma que la codificación óptima es aquella que tiene un alfabeto de cardinalidad 2 (código binario).

Es importante resaltar la importancia de una buena codificación como clave del éxito. Generalmente, la regla heurística utilizada consistente en situar cercanos en el cromosoma parámetros relacionados entre sí,

suele dar buenos resultados.

Por último, advertir que aunque normalmente la codificación es estática, en algunos casos puede convenir que el número de bits dedicados a codificar un parámetro varíe, o incluso lo que representen unas determinadas posiciones de la cadena, para, por ejemplo, adaptar automáticamente la codificación según van convergiendo los bits menos significativos de una solución.

#### **2.4.2. Función de adaptación**

Esta puede ser directamente una expresión matemática que se desea maximizar ( $F(x)$  en 2.3), el inverso de ésta, si lo que se pretende es encontrar el mínimo ( $1/F(x)$  en 2.3), estar tabulada, ser el resultado de un proceso iterativo, etc. En cualquier caso, de ella dependerá, como ya se ha visto, el grado de bondad que obtenga un determinado individuo como solución del problema. El proceso de evaluación se realiza en dos pasos:

1. Decodificación del cromosoma. Se obtiene la solución  $(x_1, \dots, x_n)$ .
2. Evaluación. Se asigna a la solución  $(x_1, \dots, x_n)$  una puntuación de acuerdo con el valor de  $y=F(x_1, \dots, x_n)$  que le corresponda. Sin embargo, puede ocurrir que no exista, o sea muy complicada de encontrar, una dependencia funcional expresable matemáticamente entre las variables  $(x_1, \dots, x_n)$  del problema con la propiedad 'y' objeto de la optimización. En este caso es preciso resolver otro problema en donde 'y' suele ser el resultado de un proceso iterativo en el que se fijan  $(x_1, \dots, x_n)$ . (Como se verá más adelante, para el problema de minimización de pérdidas de la red, "resolver otro problema" supone realizar un reparto de carga con la topología resultante de una distribución concreta de condensadores en los nudos).

Merece la pena subrayar que cualquier error cometido tanto en la codificación como en la función de adaptación que se implemente,

---

desvirtuaría la solución encontrada por el AG, en cuanto que en ellos se encierra básicamente la definición del problema.

### 2.4.3. Operadores genéticos

#### 2.4.3.1. Selección

Existe una infinidad de políticas. Enumeremos algunas de ellas:

- **Rueda de ruleta:** La selección se realiza implementando una ruleta ponderada, es decir, una rueda dividida en tantos sectores como ejemplares haya en la población, siendo el ángulo de cada sector proporcional a la bondad del individuo al que representa. Se hace girar la rueda y al pararse, una aguja fija apuntando hacia el sector afortunado, indica qué individuo es finalmente seleccionado como padre para la siguiente generación.
- **Selección de torneo:** De un subconjunto poblacional de individuos extraídos aleatoriamente, se selecciona aquél que tenga la mayor puntuación. Una ventaja respecto al método anterior es que no precisa que los valores de la función objetivo sean exclusivamente positivos (mayores que cero).
- **Basada en el rango:** Se ordena toda la población por orden de puntuación, sustituyéndose los menos aptos por descendientes del cruce de los más aptos con otros individuos de la población elegidos al azar (se superpone la población hija a la población padre).

#### 2.4.3.2. Cruce

El intercambio de información genética entre dos cromosomas se puede llevar a cabo de muchas formas, pero existen dos tipos principalmente:

- **Cruce con n puntos:** Se emparejan dos cromosomas, se cortan por n puntos elegidos al azar y se intercambian los alelos

situados entre ellos. Lo más habitual es  $n$  igual a 1 ó 2.

- **Cruce uniforme:** Se genera un patrón aleatorio de unos y ceros, y se intercambian los bits de los dos cromosomas entre las posiciones donde coincida en ambos un uno en el patrón. Otra posibilidad sería generar un número aleatorio para cada posición y si se supera un cierto valor se intercambia el bit en cuestión entre los dos cromosomas.

En ambos casos, el cruce se produce con cierta probabilidad (pcr), que suele fijarse a un valor próximo a uno.

#### **2.4.3.3. Mutación**

La decisión estriba aquí en determinar con qué frecuencia se desea que actúe este operador (pmu). En este punto, conviene advertir que un valor alto de este parámetro reduce al AG a una búsqueda aleatoria, por lo que son habituales ratios del orden de una mutación por cada mil bits procesados [2].

#### **2.4.4. Criterio de tratamiento de individuos no factibles**

Se ha visto que la codificación acota implícitamente el espacio de búsqueda. En el ejemplo de 2.3, si  $n = \log_2(b-a+1)$  resulta ser un número entero de bits, no existe ninguna combinación de  $n$  unos y ceros que codifique un entero  $x$  que no pertenezca al intervalo  $[a, b]$ . Pero en caso contrario, tanto la generación aleatoria de la población inicial como la aplicación de los operadores cruce y mutación (aunque se efectúe sobre elementos pertenecientes al dominio) podrían originar la aparición de individuos fuera del intervalo.

Además de estas restricciones, que afectan directamente al espacio de búsqueda, cualquier problema algo más complejo presentará otras del tipo:  $G(x_1, \dots, x_n) \leq A$  ó  $H(x_1, \dots, x_n) \geq B$ .

Como es obvio, hay que impedir que el AG ofrezca como solución final una que no verifique las especificaciones. Se tiene entonces dos

alternativas:

1. Prohibir la inclusión dentro de la población de estos individuos.
2. Impedir su proliferación penalizando la bondad en proporción a la desviación que presenten respecto a los límites.

La primera tiene la desventaja de que descarta toda la información contenida en el cromosoma, pudiendo haber segmentos del mismo con buenas cualidades para formar parte de la solución del problema. La segunda, menos drástica, presenta el inconveniente de trabajar con "no factibles", por lo que al parar el algoritmo tendrá que comprobarse el cumplimiento de todas las restricciones por parte de la solución alcanzada hasta ese momento.

#### **2.4.5. Criterio de inicialización**

Los AAGG tienen como peculiaridad que cada vez que se ejecutan es muy probable que aporten soluciones diferentes. Esto se debe en parte a que cada vez se inician con una población diferente, al ser ésta generada aleatoriamente.

Una estrategia habitual consiste en arrancar el algoritmo varias veces, de modo que en cada ejecución se parte con una población aleatoria pero que incorpora al mejor individuo obtenido en la anterior (se puede copiar un par de veces para evitar que se extinga accidentalmente). De esta forma, la última vez que se ejecute el algoritmo, dispone de información de mayor calidad para emprender la búsqueda hacia el óptimo.

#### **2.4.6. Criterio de parada**

En la mayoría de las aplicaciones resulta imposible conocer cuándo un AG ha alcanzado el óptimo. Por tanto, es preciso conformarse con una solución aceptable o casi-óptimo.

Por lo general se emplea como condición de terminación un número

prefijado de generaciones (`num_gen`). Ocurre en ocasiones, no obstante, que, mucho antes, la población permanece sin evolucionar a una mejor solución y con baja probabilidad de hacerlo, por estar constituida de unidades genéticamente muy semejantes. Puede entonces resultar conveniente disponer de algún indicador que avise de la pérdida de diversidad genética, para interrumpir y reiniciar (si se cree oportuno) el algoritmo.

#### **2.4.7. Parámetros de funcionamiento**

Bajo esta calificación se alude a aquellos parámetros que son propios del algoritmo genético y que son susceptibles de ser modificados con la intención de mejorar el funcionamiento del mismo. La mayoría ya han sido presentados:

1. Tamaño de la población (`num_ind`). Debe ser lo suficientemente grande como para garantizar la diversidad, por lo que tiene que estar en relación con la longitud del cromosoma. Tampoco hay que perder de vista que mayor tamaño debe implicar mayor potencia de cálculo.
2. Probabilidad de cruce (`pcr`).
3. Probabilidad de mutación (`pmu`).
4. Número máximo de generaciones (`num_gen`).

Según sea el diseño final del AG, pueden tenerse muchos otros parámetros de funcionamiento, como se desprende de las expuesto anteriormente acerca de los mecanismos de selección y cruce, criterio de parada, etc.

#### **2.4.8. Otras consideraciones**

El principal problema al que se enfrentan los AAGG consiste en la tendencia a que todos los individuos de la misma sean prácticamente idénticos. Una convergencia prematura impide que se sigan explorando

nuevas soluciones e irreversiblemente la población queda estancada en un óptimo local. Las mutaciones añaden diversidad pero no resuelven el problema.

Se examinan algunas de las prácticas destinadas a mantener la diversidad.

#### **2.4.8.1. Escalado de la función objetivo**

El escalado se implementa para proporcionar niveles adecuados de competición a lo largo de la simulación: Las primeras fases de la evolución están corrientemente dominadas por unos pocos super-individuos que apenas dan oportunidad a individuos peor dotados que ellos, con lo que la diversidad tiende a decaer muy rápidamente. Por contra, cuando la población tiende a uniformizarse, las "fuerzas" se hallan equiparadas y, en consecuencia, la simulación se comporta erráticamente (sólo a expensas de las mutaciones que se originen).

Con el escalado se disminuyen o acentúan las diferencias de bondad entre los individuos, según se desee.

Existen varios métodos [2]. Se presentan los dos siguientes:

1. Escalado lineal. Sea  $F$  la función objetivo; entonces, la función escalada es de la forma:  $F' = aF + b$ . Con  $a$  fijo, subir  $b$  implica disminuir las diferencias relativas. Con  $b$  fijo, aumentar  $a$  supone acrecentarlas.
2. Escalado exponencial. En este caso,  $F' = F^k$ . Un  $k$  mayor que uno agranda las diferencias, mientras que un  $k$  menor que uno, las reduce.

#### **2.4.8.2. Compartir la bondad**

En la bibliografía se denota por el término inglés "sharing" [4]. Consiste en dividir la bondad de cada individuo por un valor que está en función de su parecido (ya sea del fenotipo o del genotipo) con el

resto de la población: Para un individuo dado,  $X_j$ , se define  $d_{jk}$  como una medida de la "distancia" (diferencia fenotípica o cromosómica) con cada uno de los restantes elementos poblacionales  $X_k$ . Se considera una función  $S(d_{jk})$ , de tal forma que toma un valor tanto más próximo a uno cuanto menor sea la distancia entre  $X_j$  y  $X_k$ , y un valor tendiendo a cero en el caso contrario. Entonces, la bondad corregida de  $X_j$ ,  $F_s(X_j)$ , se expresa como sigue:

$$F_s(X_j) = \frac{F(X_j)}{\sum_{k=1}^{num\_ind} S(d_{jk})}$$

Empleando en la selección  $F_s$  en lugar de  $F$ , se favorece la aparición de representantes de los distintos extremos locales de la función objetivo, en lugar de converger la población hacia uno de ellos. La implementación de este operador conlleva una carga de cálculo adicional: Hay que evaluar cuán parecido es cada individuo al resto de la población.

#### 2.4.8.3. Apiñamiento o "crowding"

El crowding [8] persigue lo mismo que el sharing. En primer lugar se determina aleatoriamente un subconjunto poblacional,  $S$ , al que se le aplicará los operadores genéticos. Cuando nace una pareja de nuevos individuos, llamémosles  $H1$  y  $H2$ , como resultado de la selección de dos elementos de  $S$ , su cruce y eventual mutación, el siguiente paso es determinar cuáles dos individuos deben morir para dejarles hueco en la actual población (se superpone la generación hija a la generación padre). Tanto para  $H1$  como para  $H2$ , la selección del que va a ser reemplazado se hace así:

1. Se selecciona aleatoriamente un subconjunto  $C$  de individuos de toda la población.
2. De entre los miembros de  $C$  se toma el peor y se compara (bit a

bit) con H1 (ó H2), cuantificando su parecido.

3. Se repite el proceso un número de veces igual a CF ("crowding factor").
4. El individuo que muere es aquél que resulta más parecido a H1 (ó H2)

CF suele tomarse igual a 2 ó 3.

#### **2.4.8.4. Elitismo**

Un modelo elitista [8] sería aquél que preservara de la extinción a los mejores individuos de la última generación obtenida, que pasan directamente (sin alteración alguna) a formar parte de la población de descendientes.