

Estrategias Evolutivas : La Versión Alemana del Algoritmo Genético (Parte II)

Arturo Hernández Aguirre
Bill P. Buckles
Carlos A. Coello Coello

En la segunda y última parte de este artículo hablaremos sobre las estrategias evolutivas de varios miembros, que son realmente las que se usan más en la práctica; veremos en más detalle el funcionamiento del algoritmo genético para este tipo de estrategia evolutiva, así como la forma en que se implementan sus operadores de mutación y cruza. Finalmente, se proporcionará un ejemplo de su uso y se hará referencia a algún software disponible en el Internet que puede usarse para experimentar con esta técnica evolutiva.

Las estrategias multimiembro $EE-(\mu+\lambda)$ y $EE-(\mu,\lambda)$

La motivación para extender el modelo $EE-(\mu+1)$ ¹, a los modelos $EE-(\mu+\lambda)$ y $EE-(\mu,\lambda)$ de varios miembros, fue doble : primero, aprovechar las arquitecturas en paralelo, y segundo, permitir la auto-adaptación de los parámetros estratégicos, es decir, las desviaciones estándar de las mutaciones. En estos dos modelos ya no se usa una regla externa determinística (“éxito de 1/5”) para modificar las desviaciones estándar, sino que ahora éstas forman parte de la información de todos y cada uno de los individuos de la población y por lo tanto están sujetas a las operaciones de mutación y recombinación. De esta forma la modificación de las desviaciones estándar se convierte en un proceso auto-adaptable que se desarrolla en el ambiente de evolución artificial del algoritmo. Se espera que aquellos individuos que mejor adapten sus parámetros estratégicos tengan mejor valor de aptitud y que por lo tanto tengan alta probabilidad de ser seleccionados para poblar la siguiente generación. Eventualmente estos individuos dominarán en la población y la auto-adaptación hará que surjan mejores parámetros estratégicos. La diferencia entre los modelos $EE-(\mu+\lambda)$ y $EE-(\mu,\lambda)$ estriba en la forma en la que los padres de una nueva generación son seleccionados. En la $EE-(\mu+\lambda)$ mostrada en la Figura 1, los μ padres junto con sus λ hijos son reunidos en un solo conjunto del cual se seleccionan los μ individuos más aptos para formar la siguiente generación (nótese que cuando μ y λ tienen un valor de 1, el modelo se reduce a la versión más simple del algoritmo, llamada $EE-(1+1)$).

¹ En la cual μ padres (donde $\mu > 1$) pueden participar en la generación del único hijo de una generación.

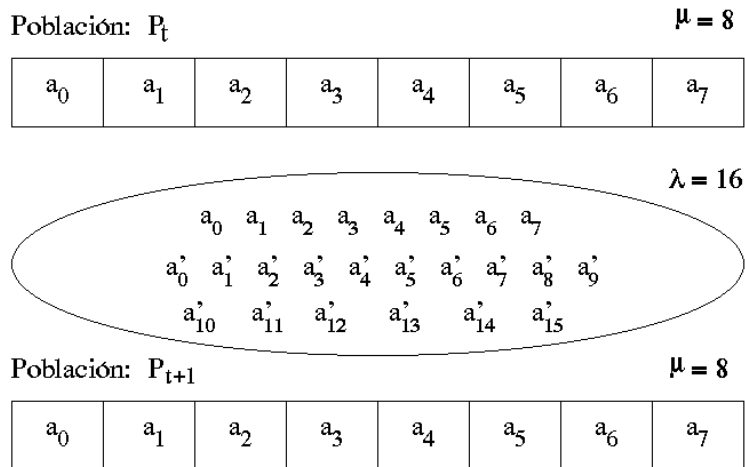


Figura 1. La estrategia EE-($\mu+\lambda$)

En la EE-(μ,λ) que se ilustra en la Figura 2 ($\lambda > \mu \geq 1$), los μ individuos de la nueva generación son seleccionados exclusivamente de los λ hijos producidos, sin importar si son o no mejores que sus padres.

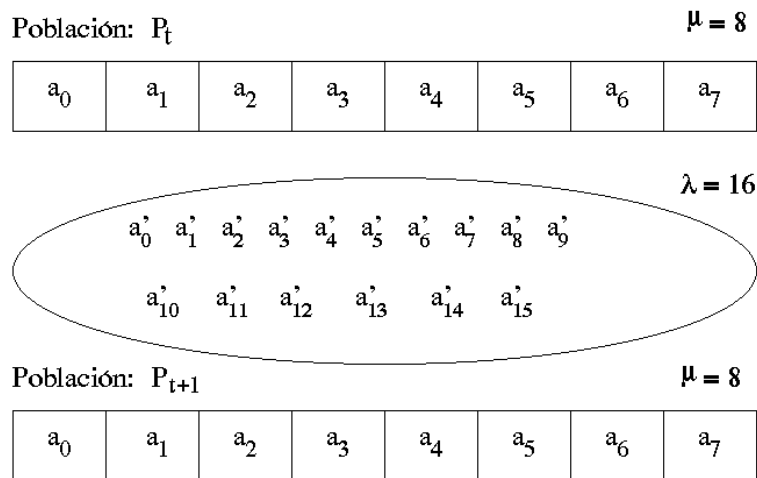


Figura 2. La estrategia EE-(μ,λ)

La mutación en las EEs realiza una especie de búsqueda escalando la colina (*hill climbing*), cuando se combina con la selección. En los modelos EE-($\mu+\lambda$) y EE-(μ,λ) cada variable objetivo usa su propia desviación estándar asignada para producir mutaciones o alteraciones a sus valores nominales (es decir, no correlacionados) que producen desplazamientos a lo largo de los ejes del sistema de coordenadas. Sin embargo, la mejor dirección de búsqueda para este desplazamiento sería en la dirección del óptimo, el cual es muy poco probable que se encuentre alineado con los ejes coordenados, como se ve en la Figura 3. Esto significa que la trayectoria de

búsqueda sigue un camino en forma de zigzag más o menos perpendicular al vector en la dirección del óptimo.

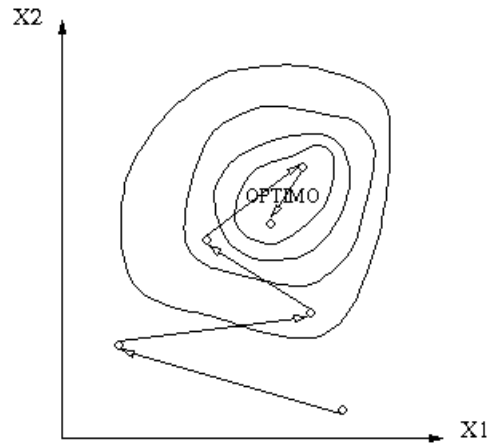


Figura 3. Búsqueda del óptimo

Buscando evitar esta reducción en la velocidad de convergencia, Schwefel modificó el operador de mutación para que fuera capaz de realizar mutaciones correlacionadas, lo cual agregó una variable estratégica más, el ángulo θ , o dirección preferente de movimiento. Con el uso de mutaciones correlacionadas, los movimientos más favorecidos serán aquellos que sigan la dirección del vector que nos lleve al máximo de la función, y esto se logra minimizando la ruta en zigzag que tienen los organismos en el espacio de búsqueda cuando las mutaciones son independientes. Una mutación correlacionada se obtiene haciendo girar una mutación no correlacionada un ángulo θ sobre el plano². El ángulo θ es proporcional a la covarianza c_{ij} que existe entre σ_i y σ_j de acuerdo a la relación :

$$\text{Tan}(2\theta_{ij}) = \frac{2c_{ij}}{(\sigma_i^2 - \sigma_j^2)}$$

Las combinaciones de $\langle \sigma_1, \sigma_2, \dots, \sigma_k \rangle$ desviaciones estándar producen una matriz de covarianzas, C , con k^2 elementos. Para $k = 2$ variables objetivo la matriz es la siguiente :

$$C = \begin{vmatrix} \sigma_1\sigma_1 & \sigma_1\sigma_2 \\ \sigma_2\sigma_1 & \sigma_2\sigma_2 \end{vmatrix}$$

² O un hiperplano en el caso de k dimensiones.

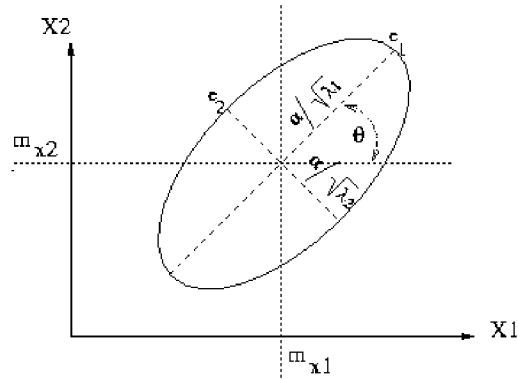


Figura 4. Representación geométrica de un organismo de las EEs

La matriz de covarianzas, C , define el lugar geométrico de los puntos de una elipsoide con centro en (m_{x1}, m_{x2}) , que satisfacen la ecuación:

$$\alpha^2 = (X - m)^T C^{-1} (X - m)$$

donde α es una constante real positiva, \mathbf{X} son valores reales sobre los ejes X_1 y X_2 que satisfacen la ecuación y m son las coordenadas del centro de la elipse. El centro de la elipse es la coordenada indicada por la variable objetivo. Los eigenvalores $\lambda_1, \lambda_2, \dots, \lambda_k$, de la matriz inversa de covarianzas C^{-1} , definen la longitud de los ejes e_1, e_2, \dots, e_k , y en consecuencia, son los que dan el tamaño de la elipsoide. Los eigenvectores de C^{-1} definen a su vez la orientación de los ejes, como se muestra en la Figura 4.

Una vez que se cuenta con las *variables objetivo* y las *variables de control o estratégicas*, las técnicas EE- $(\mu+\lambda)$ y EE- (μ,λ) se definen como estrategias aleatorias de búsqueda con parámetros de control auto-adaptables, donde el tamaño del paso o mutación está controlado por σ y la dirección preferente de movimiento por θ .

Cada organismo se constituye por un vector de variables objetivo $\mathbf{X}=\langle x_1, x_2, \dots, x_k \rangle$, y un vector de variables estratégicas que consta de dos partes: el vector de desviaciones estándar $\sigma=\langle \sigma_1, \sigma_2, \dots, \sigma_k \rangle$, y el vector de direcciones de movimiento $\theta=\langle \theta_1, \theta_2, \dots, \theta_m \rangle$. La Figura 5 muestra a ciertos organismos de la población buscando alcanzar el máximo como un punto en la superficie de respuesta $f(x, y)$.

La estructura de cada organismo consta de un vector de dos variables objetivo $\mathbf{X}=\langle x_1, x_2 \rangle$, y de dos vectores de variables estratégicas: el vector de desviaciones estándar $\sigma=\langle \sigma_1, \sigma_2 \rangle$ y el vector de direcciones $\theta = \langle \theta_i \rangle$. El centro de la elipse está indicado por las variables objetivo; la longitud y dirección de los ejes de las elipses son proporcionales a la magnitud y signo de las correspondientes $\langle \sigma_1, \sigma_2 \rangle$, y finalmente, el ángulo de orientación de cada elipse es $\langle \theta_i \rangle$.

Figura 5. Tres individuos en una superficie de respuesta

En el caso más general, el vector \mathbf{X} de variables objetivo y el vector $\boldsymbol{\sigma}$ de desviaciones estándar tienen el mismo número k de elementos. El número m de elementos en el vector $\boldsymbol{\theta}$ es $k(k-1)/2$. Esto es, cada variable objetivo x_j tiene su propia desviación estándar σ_j , y para cada pareja $\sigma_i\sigma_j$ existe una θ_m correspondiente. Bäck [1] explica que este caso general no es del todo necesario; es decir, algunas variables objetivo pueden compartir determinada σ_j y determinadas parejas de $\sigma_i\sigma_j$ pueden compartir a un ángulo θ_k con la finalidad de disminuir el tiempo de cómputo empleado en calcular las rotaciones. La relación entre las variables estratégicas y la velocidad de convergencia permanece como un área abierta de investigación.

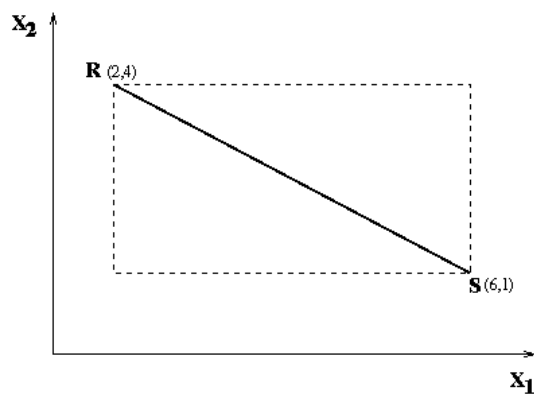


Figura 6. Recombinación en las Estrategias Evolutivas

El algoritmo $EE-(\mu \ddagger \lambda)^3$

³ El uso de \ddagger implica que este algoritmo es suficientemente general como para incluir la $EE-(\mu+\lambda)$ y la $EE-(\mu,\lambda)$.

Como hemos visto, el paradigma de las EEs ha evolucionado durante los últimos 30 años desde una forma relativamente simple hasta la concepción actual que es bastante elaborada. A continuación se describe de manera más detallada el uso y diversas variantes de los operadores de recombinación, selección, mutación y aptitud que forman el corazón del algoritmo, procediendo de acuerdo al orden en que se aplican en el siguiente esquema conceptual de una EE [1][2][3][4][5][6][7] :

```

1. [Inicializar población]
   t := 0;
   P0(μ) := {a0(1) <X, σ, θ>, a0(s) <X, σ, θ>, ..., a0(μ) <X, σ, θ>}
2. [Evaluar aptitud]
   P0(μ) := {φ(a0(1)), φ(a0(2)), ..., φ(a0(μ))}
3. [Comenzar ciclo ]
   For (t:=1 to max_iter step 1) do
     Recombinación: producir λ hijos con μ padres
     P't(λ) <X, σ, θ> := Rec(Pt(μ) <X, σ, θ>)
     Mutación: producir una mutación a cada hijo
     P''t(λ) <X, σ, θ> := Mut(P't(λ) <X, σ, θ>)
     Aptitud: calcular la aptitud de cada hijo
     P''t(λ) := φ( P''t(λ))
     Selección: seleccionar los μ individuos de la siguiente generación
     Pt+1(μ) <X, σ, θ> := Sel(P''t(λ) <X, σ, θ>) /* si EE-(μ+λ) */
     Pt+1(μ) <X, σ, θ> := Sel(P''t(λ) <X, σ, θ>) ∪ Pt(μ) <X, σ, θ> /* si EE-(μ † λ) */
   Endfor

```

Recombinación

La recombinación es el operador por medio del cual la información transportada por los individuos de una población es compartida para crear nuevos individuos o hijos. En la EE existen dos estrategias básicas para la recombinación: la *sexual*, en la que dos padres son elegidos al azar para todos los parámetros <X, σ, θ>, y la *panmítica*, en la que un padre elegido al azar se recombina con otro elegido de la misma forma por cada parámetro en <X, σ, θ>. Las variantes conocidas para ambas estrategias son iguales pero aquí ilustramos sólo la estrategia sexual mediante dos organismos hipotéticos de la población, efectuando la recombinación únicamente sobre el vector X de dos variables objetivo.

Usando la estrategia sexual, dos individuos R=<2,4> y S=<6,1>, producen hijos de la siguiente forma:

- *no hay recombinación*: hijo igual al padre R.
- *discreta*: cualquiera de cuatro hijos posibles: <2,4>, <6,1>, <2,1>, <6,4>; donde el valor que toma cada hijo proviene de un padre elegido al azar.
- *intermedia*: el hijo es <2+(6-2)/2, 4+(1-4)/2> =<4,2.5>
- *generalizada intermedia*: el hijo es <2 + η(6-2), 4 + η(1-4)>, donde η es un número aleatorio en el rango [0..1]

En la Figura 6 se ilustra este ejemplo. Los hijos de la recombinación *discreta* se encuentran en las cuatro esquinas del rectángulo. La recombinación *intermedia* produce un hijo a la mitad de la línea

sólida y la recombinación *generalizada intermedia* produce un hijo en cualquier punto a lo largo de la línea sólida. Otras reglas de recombinación, como la *panmítica generalizada*, producen hijos en cualquier zona dentro del rectángulo.

La recombinación de las variables de control puede emplear una regla diferente a la usada en las variables objetivo; esto es, una estrategia de recombinación puede emplear hasta tres reglas diferentes: una para el vector \mathbf{X} , otra para el vector $\boldsymbol{\sigma}$ y otra más para el vector $\boldsymbol{\theta}$.

Mutación

La mutación consiste en alterar una variable objetivo o de control mediante un número aleatorio. Son estas mutaciones las que esencialmente efectúan la búsqueda. En las EEs, las mutaciones se realizan primero en las variables de control y después en las variables objetivo. Las reglas para los vectores $\boldsymbol{\sigma}$ y $\boldsymbol{\theta}$ son las siguientes:

$$\sigma'_i = \sigma_i \exp(\tau' N(0,1) + \tau N_i(0,1)) \quad (1)$$

$$\theta'_j = \theta_j + \beta N_j(0,1) \quad (2)$$

Donde $N_i(0,1)$ y $N_j(0,1)$ son números aleatorios con distribución normal (media = 0 y esperanza = 1) que se generan para cada σ_i y θ_j , mientras que $N(0,1)$ es otro número aleatorio que permanece fijo durante la mutación. τ , τ' y β son constantes funcionalmente equivalentes al "factor de aprendizaje" empleado en las redes neuronales. Los valores típicos reportados por Schwefel [22] dependen del número k de variables objetivo, y son :

$$\tau = 1 / (4 \cdot k^{1/4}), \quad \tau' = 1 / (2 \cdot k)^{1/2} \quad \text{y} \quad \beta \approx 0.0873.$$

Una vez que se han mutado los vectores $\boldsymbol{\sigma}$ y $\boldsymbol{\theta}$, se aplica la regla para mutar el vector \mathbf{X} de variables objetivo el cual se basa en alterar cada variable por medio de mutaciones correlacionadas que se generan empleando el vector de ángulos $\boldsymbol{\theta}$. La regla de mutación correlacionada se expresa como sigue:

$$\mathbf{x}' = \mathbf{x} + N(0, C(\boldsymbol{\sigma}, \boldsymbol{\theta})) \quad (3)$$

En esta ecuación, el término $N(0, C(\boldsymbol{\sigma}, \boldsymbol{\theta}))$ implica que el número aleatorio se genera empleando distribución normal con media cero y varianza contenida en la matriz de covarianzas $C(\boldsymbol{\sigma}, \boldsymbol{\theta})$. El proceso de mutación se expresa en el siguiente algoritmo⁴:

hijo ← mutación(organismo)

1. $x[1 : k] = \mathbf{X}$
2. $\sigma[1 : k] = \boldsymbol{\sigma}$
3. $m = k(k-1)/2$

⁴ Recuerde que cada organismo lo hemos representado como una concatenación de tres vectores, en la forma $\langle \mathbf{X}, \boldsymbol{\sigma}, \boldsymbol{\theta} \rangle$, y que el número de variables objetivo de cada organismo es k (véase la Figura 5).

```

4.    $\theta[1 : m] = \Theta$ 
5.    $r = \text{random}()$ 
6.   for  $i = 1$  to  $k$ 
7.        $s = \text{random}()$ 
8.        $\sigma[i] = \sigma[i] \exp(\tau \cdot r + \tau \cdot s)$ 
9.   Endfor
10.  for  $j = 1$  to  $k$ 
11.       $s = \text{random}()$ 
12.       $\theta[j] = \theta[j] + \beta \cdot s$ 
13.  Endfor
14.  for  $i = 1$  to  $k$ 
15.       $\Delta x[i] = \sigma[i] \cdot \text{random}()$ 
16.  Endfor
17.  for  $k = m$  downto  $1$ 
18.       $(i, j) = \text{indices}(k)$ 
19.       $\Delta x[i] = \Delta x[i] \cos \theta[k] - \Delta x[j] \sin \theta[k]$ 
20.       $\Delta x[j] = \Delta x[i] \sin \theta[k] + \Delta x[j] \cos \theta[k]$ 
21.  Endfor
22.  for  $i = 1$  to  $k$ 
23.       $x[i] = x[i] + \Delta x[i]$ 
24.  Endfor
25.  hijo =  $\langle x[1 : k], \sigma[1 : k], \theta[1 : m] \rangle$ 

```

La función *random()* retorna un número aleatorio con distribución estándar normal. La función *indices()* proporciona las combinaciones correctas de índices que permiten seleccionar para cada θ_k sus valores correspondientes de σ_i y σ_k . Las líneas 6-9 realizan el cálculo de la ecuación (1) y las líneas 10-13 el de la ecuación (2). En las líneas 14-16 se produce una mutación no correlacionada, y posteriormente se rota en las líneas 17-21 por medio de los ángulos θ para convertirla en correlacionada. Nótese el orden de rotación comenzando por el último ángulo, como lo explica Schwefel en su libro [3]. En las líneas 22-25 se realiza finalmente la mutación de las variables objetivo.

Aptitud.

La aptitud es una medida de la calidad de la solución que cada individuo de la población contiene en su estructura. Se calcula usualmente por medio de una función que recibe como parámetros el vector de variables objetivo \mathbf{X} y devuelve una cantidad escalar de la siguiente forma:

$$\text{aptitud} \leftarrow \Phi(\mathbf{X})$$

Usualmente, Φ es la función que se está optimizando.

Selección.

El operador de selección es probablemente el más sencillo de implementar. Su función consiste en seleccionar los λ individuos con más alto valor de aptitud que van a poblar la siguiente generación.

Un breve ejemplo

A continuación se ilustra una aplicación de las EEs en la maximización de una función de dos variables independientes⁵:

$$f(x, y) = x \exp(-x^2 - y^2)$$

que se ilustra en la Figura 7.

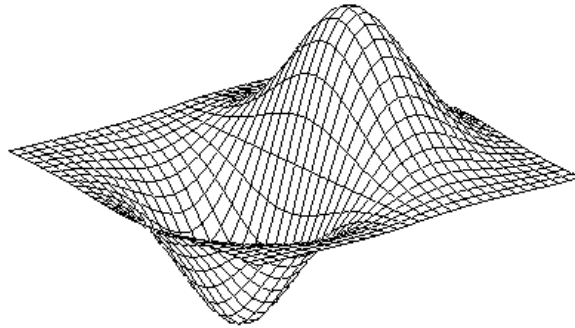


Figura 7. La función del caso de estudio, $f(x, y) = x \exp(-x^2 - y^2)$

Un organismo en este ejercicio tiene la forma usual $\langle \mathbf{X}, \boldsymbol{\sigma}, \boldsymbol{\theta} \rangle$, con elementos $\langle x_1, x_2; \sigma_1, \sigma_2; \theta_1 \rangle$. En este problema se utilizó una estrategia EE-(5,10) con dos diferentes reglas de recombinación: sexual intermedia para las variables objetivo x_1 y x_2 , y sexual discreta para las variables de control σ_1 , σ_2 y θ_1 . La mutación se realizó usando el algoritmo descrito anteriormente.

⁵ Si el lector desea obtener el código fuente en Matlab para reproducir este experimento, solicítelo al primer autor a la dirección electrónica : hernanda@eecs.tulane.edu.

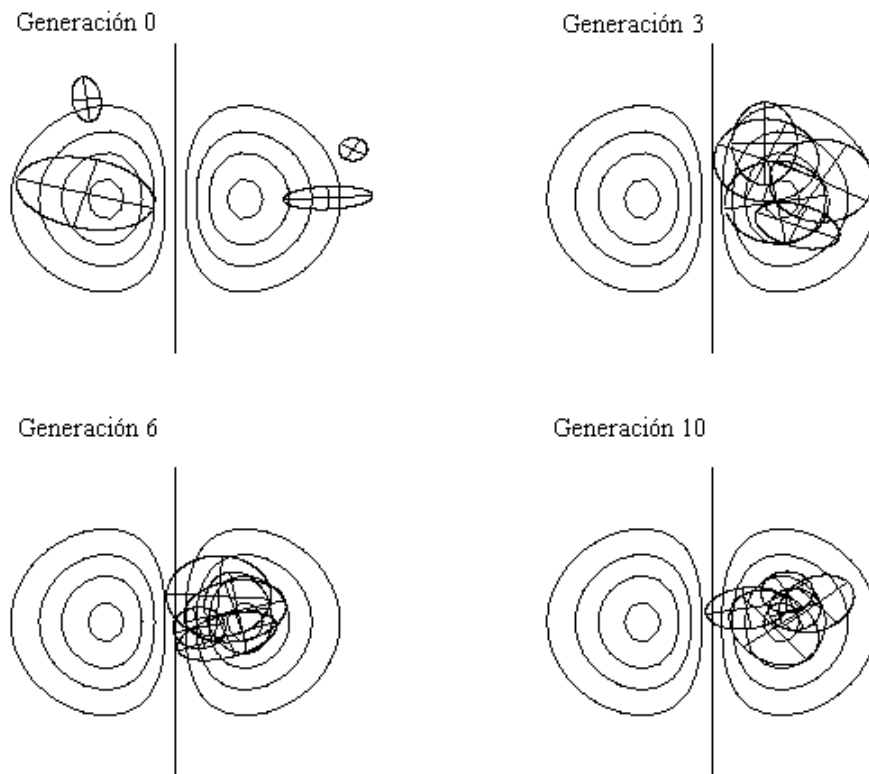


Figura 8. Progresión de los individuos de la EE hacia el óptimo

La Figura 8 muestra una secuencia de búsqueda típica de las EEs. En la parte superior izquierda se encuentra la población inicial. Las generaciones 3 y 6 se muestran en la parte superior derecha e inferior izquierda, respectivamente. Como se puede apreciar, existe una fuerte orientación y progresión de los organismos hacia el máximo. La última generación se muestra en la parte inferior derecha de la figura, donde se aprecia que los individuos se han conglomerado alrededor del máximo. También es notorio que las elipsoides son más pequeñas y no están alargadas ya que las diferencias entre las σ 's se han atenuado y han adquirido valores más uniformes. Esto es debido a que las variables de control σ 's, son auto-adaptables en el ambiente de evolución artificial inducido por la EE. Cuando el óptimo es alcanzado, las σ 's se auto-reducen al mínimo para así, a su vez, reducir la mutación. Una menor mutación se traduce en desplazamientos más pequeños en la vecindad del óptimo. La presencia de organismos de forma más bien redonda que alargada, denota que ya no existe una dirección preferente de movimiento, y que esencialmente, la búsqueda ha terminado.

Ambientes de Programación

Existen en el Internet varias plataformas para implementar estrategias evolutivas en una computadora. De entre ellas, quizás la más conocida sea **evoC**, que fue desarrollada en el Departamento de Biónica y Evolución de la Universidad Técnica de Berlín. Este programa está escrito en ANSI C y proporciona un ambiente de programación bastante amigable para la

implementación de diversas variantes de las estrategias evolutivas en varias plataformas (SunOS, IRIX, HP-UX, Linux y MS-DOS). Aunque el software es gratuito, no es de dominio público (tiene una licencia similar a la del software GNU), y puede conseguirse (incluyendo la documentación) mediante 'ftp' *anonymous* del nodo: **ftp-bionik.fb10.tu-berlin.de** en el directorio **/pub/software/evoC**.

También existe una caja de herramientas para el desarrollo de estrategias evolutivas que corre bajo *Matlab*, y que proporciona una interfaz gráfica amigable así como la posibilidad de trabajar con problemas de uno o varios objetivos. Este software puede conseguirse por medio del Dr. To Thanh Binh del Instituto de Automatización de la Universidad de Magdeburg, en Alemania (su dirección electrónica es **binh@infaut.et.uni-magdeburg.de**).

Si lo que le interesa es software más sencillo y del cual pueda obtener el código fuente, existen 2 buenas opciones:

- **LICE** (*Linear Cellular Evolution strategy*) : Es un programa para la optimización de parámetros basado en las estrategias evolutivas que tiene un esquema de selección local que impide que haya estancamiento prematuro. El código puede obtenerse contactando a su autor, Joachim Sprave (**joe@LS11.informatik.uni-dortmund.de**) en el Departamento de Ciencias de la Computación y Análisis de Sistemas de la Universidad Dortmund, en Alemania.

- **SES** (*Simple Evolution Strategy*) : Es un programa de optimización de parámetros basado en las estrategias evolutivas que no implementa las mutaciones correlacionadas, ni la recombinación global. El código puede obtenerse contactando también a Joachim Sprave (el autor de LICE).

Conclusiones

A la luz de lo expuesto en este artículo, es importante recordar que las estrategias evolutivas (EEs) se enfocan a resolver problemas de optimización de funciones reales, mientras que los algoritmos genéticos (AGs) parecen ser mucho más independientes del dominio, si bien su desempeño en la optimización de funciones reales suele ser más pobre que el de las estrategias evolutivas. Asimismo, en las EEs, las reglas de recombinación, mutación y selección se encuentran altamente estructuradas, a comparación de los mecanismos equivalentes en los AGs, que suelen ser más aleatorios.

Otro punto interesante acerca de las estrategias evolutivas es que evolucionan simultáneamente una solución y las variables de control que la llevan a encontrar la misma, y ésta es precisamente una de sus diferencias principales con respecto a los algoritmos genéticos.

El hecho de que una estrategia evolutiva contenga en un mismo individuo a su vector de soluciones y sus variables de control, da pie a muchas interrogantes. En primera instancia, es razonable preguntarse si esta doble tarea no resulta excesiva para nuestro mecanismo de selección. En segundo término, cabe preguntarse: ¿qué debemos seleccionar, un organismo cerca de la solución pero mal orientado, o uno alejado de la solución pero correctamente orientado? Las respuestas a estas preguntas siguen siendo motivo de debate e investigación, y generalmente se responden en el contexto determinado de una aplicación en vez de intentar hacerlo de manera general. De tal forma, este problema de representación de las EEs constituye su talón de Aquiles en la práctica, haciendo las veces de los parámetros de ajuste de un AG.

Otro punto oscuro en torno al uso de las estrategias evolutivas cómo definir una función de aptitud que efectivamente nos lleve a la zona donde reside el óptimo. Esto generalmente no es fácil en los problemas hacia los cuales está dirigida esta técnica, y requiere normalmente de bastante experimentación. Este problema se presenta también en los algoritmos genéticos en donde, además, la codificación inapropiada de las variables dentro de la cadena cromosómica puede hacer que la técnica falle, aunque se cuente con una función de aptitud apropiada.

Aunque el uso de las estrategias evolutivas requiere un conocimiento más profundo de matemáticas (particularmente de estadística), esta técnica es evidentemente más poderosa que el algoritmo genético para optimizar funciones reales de gran complejidad en las que los métodos tradicionales parecen no producir resultados satisfactorios. De tal forma, puede decirse que los dominios de ambas técnicas evolutivas, más que traslaparse, se complementan, y por ello resulta importante conocer el alcance de cada una para poder elegir la más apropiada en la aplicación específica que nos interese.

Referencias

[1] Bäck, T., "Evolutionary Algorithms in Theory and Practice", Oxford University Press, 1996.

[2] Bäck, T., "Evolution Strategies: An Alternative Evolutionary Algorithm", en Alliot, J. M., Lutton E. (eds.), *Artificial Evolution*, LNCS 1063, Springer Verlag, 1996.

[3] Schwefel, Hans-Paul, "Evolution and Optimum Seeking", John Wiley and Sons, 1995.

[4] Schwefel, Hans-Paul. "Numerical Optimization of Computer Models", John Wiley & Sons, Great Britain, 1981.

[5] Schwefel, Hans-Paul. "Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie", vol. 26 de *Interdisciplinary Systems Research*, Birkhäuser, Basel, 1977.

[6] Bäck, T., Schwefel, H. P., "Evolution Strategies I: Variants and their computational implementation", en Périaux J., Winter G. (eds.), *Genetic Algorithms in Engineering and Computer Science*, John Wiley and Sons, 1995.

[7] Schwefel, H. P. & Bäck T., "Evolution Strategies II: Variants and their computational implementation", en Périaux J., Winter G., (eds.), *Genetic Algorithms in Engineering and Computer Science*, John Wiley and Sons, 1995.