

## OPTIMIZACIÓN COMPLEMENTOS - IV

CESÁREO RAIMÚNDEZ

### 1. EVOLUCIÓN - PARADIGMA PARA ENFOCAR PROBLEMAS COMPLEJOS

Basados en la ley enunciada por Darwin, surgen diversos procedimientos de búsqueda de máximos/mínimos, en condiciones que difícilmente podrían tratarse al sabor de las técnicas clásicas. La supervivencia de los más aptos y la continuación de su progenie, es la idea básica que soporta las técnicas evolutivas. Los algoritmos utilizados son una realización sumamente estilizada que intenta simular el proceso evolutivo. Para tanto se necesitan elementos como:

**Individuo:** Objeto sensible al proceso evolutivo. Posee elementos propios de su singularidad. (Cromosoma)

**Población:** Conjunto de Individuos que evolucionan e inter actúan entre sí.

**Ambiente:** Elemento externo que exprime las características probabilísticas de la interacción individual.

**Adecuación:** Sensor que determina cuan adecuado con relación al ambiente, está un individuo.

### 2. VERSIÓN BINARIA

Veremos cómo aplicar este paradigma a la resolución de un problema de determinación del máximo de la función

$$f(x) = e^{-8(x_1+0.5)^2(x_2-1)^2} + e^{-4(x_1-0.5)^2(x_2+1)^2}$$

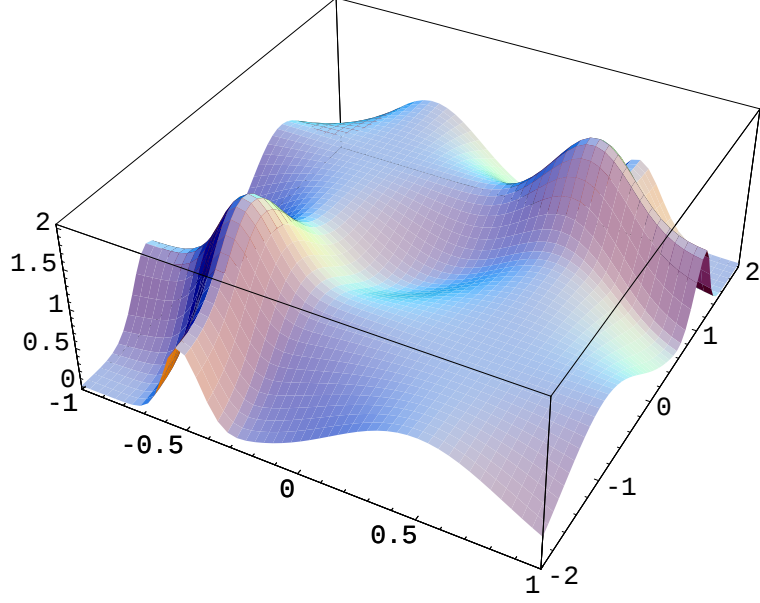
para  $x \in \mathbb{R}^2$ .

Definiremos el Individuo como teniendo un atributo básico: su Cromosoma. El Cromosoma se entenderá como el conjunto de elementos que lo hacen singular y a través del que puede transmitir sus características a su progenie. En el caso espacial que nos anima, el cromosoma estará formado por una cadena de elementos atómicos escogidos sobre un alfabeto determinado. En este caso utilizaremos como alfabeto la base binaria  $\{0, 1\}$ . Cada individuo está caracterizado por la posición que ocupa en el espacio en que está definido. Este espacio debe poder codificarse utilizando el alfabeto adoptado. Así un Cromosoma típico del  $i$ -ésimo individuo tendrá el aspecto

$$C_i = \{ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ \cdots \ 1 \ 0 \ 0 \}, \ i = 1, \dots, n_I$$

donde  $n_I$  es el número de individuos que forman una población. Consideraremos que todos individuos poseen cromosomas con cadenas de misma longitud que denotaremos como

$$n_C = |C|$$

FIGURA 1. Aspecto de la superficie  $f(x)$ 

Sea el problema de determinar el máximo de  $f(x)$  en el conjunto  $\Omega$

$$\Omega = \{x_1^- \leq x_1 \leq x_1^+, x_2^- \leq x_2 \leq x_2^+\} \subset \mathbb{R}^2$$

Hagamos ahora una discretización del espacio de búsqueda de acuerdo con

$$(2^{n_1} - 1)\Delta_1 = x_1^+ - x_1^-, (2^{n_2} - 1)\Delta_2 = x_2^+ - x_2^-$$

de esta forma, la discretización de base  $\Delta_1, \Delta_2$  podrá ser descrita a través de cadenas de longitud  $n_1, n_2$  construidas sobre el alfabeto binario. Esta aplicación que relaciona  $x$  en el espacio  $\mathbb{R}^2$  con  $\hat{x}$  en el espacio reticulado  $\hat{\Omega}$ , puede ajustarse de forma a expresar el espacio continuo dentro de una aproximación estipulada por el tamaño de  $\Delta_1, \Delta_2$ .

### 2.1. Ejemplo. Sean

$$x_1^- = -1, x_1^+ = 1, x_2^- = -2, x_2^+ = 2$$

adoptando  $n_1 = 6, n_2 = 7$  tendremos  $\Delta_1 = 0.031746, \Delta_2 = 0.0314961$ . El este sistema reticulado, en las nuevas variables

$$\hat{x}_1 = x_1^- + \Delta_1 D(b_1), \hat{x}_2 = x_2^- + \Delta_2 D(b_2)$$

donde  $b_1, b_2$  son secuencias binarias construidas sobre el alfabeto  $\{0, 1\}$  con longitudes máximas  $n_1, n_2$  respectivamente, y el operador  $D(\cdot)$  es el que convierte el número de base 2 a base 10. De esta guisa, el punto  $\hat{x} = (0.174603, 1.30709)$  tiene su representación codificada en el alfabeto binario como

$$\begin{aligned} \hat{x}_1 &= 0.174603 \rightarrow \{1 \ 0 \ 0 \ 1 \ 0 \ 1\} \\ \hat{x}_2 &= 1.30709 \rightarrow \{1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1\} \end{aligned}$$

### 3. CARACTERIZACIÓN DEL INDIVIDUO

El contenido que caracteriza el individuo en el subconjunto  $\hat{\Omega}$  son sus coordenadas cartesianas. Tomaremos esta información como el contenido de su cromosoma. Consideremos su cromosoma formado por las cadenas de bits asociadas sus coordenadas en ambas variables, colocadas en un orden pre establecido como sea el ordenado por el índice de la variable considerada.

$$C_1 = \{ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \}$$

Una población de 10 individuos generada aleatoriamente sobre las secuencias binarias de largura  $n_1 + n_2 = 13$  y presentada en una tabla, tendría el aspecto

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0.4920634920634921 & -1.968503937007874 \\ -0.7777777777777778 & -1.5590551181102361 \\ 0.9682539682539681 & 1.1496062992125982 \\ -0.3650793650793651 & -0.20472440944881898 \\ 0.26984126984126977 & 0.04724409448818889 \\ -0.39682539682539686 & -1.4645669291338583 \\ 0.7142857142857142 & 0.8346456692913384 \\ -0.4920634920634921 & -0.8661417322834646 \\ -0.5555555555555556 & -0.9606299212598426 \\ -0.6507936507936508 & 0.393700787401575 \end{bmatrix}$$

Las secuencias binarias decodificadas generarían pares de coordenadas que estarían ubicadas en  $\hat{\Omega}$ , conforme mostrado arriba.

### 4. ADECUACIÓN

La adecuación del individuo al ambiente se mide a través del valor que su contenido (coordenadas en  $\hat{\Omega}$ ) asume de acuerdo con la función objetivo. Así en el ejemplo actual tendríamos:

$$f \begin{pmatrix} 0.4920634920634921 & -1.968503937007874 \\ -0.7777777777777778 & -1.5590551181102361 \\ 0.9682539682539681 & 1.1496062992125982 \\ -0.3650793650793651 & -0.20472440944881898 \\ 0.26984126984126977 & 0.04724409448818889 \\ -0.39682539682539686 & -1.4645669291338583 \\ 0.7142857142857142 & 0.8346456692913384 \\ -0.4920634920634921 & -0.8661417322834646 \\ -0.5555555555555556 & -0.9606299212598426 \\ -0.6507936507936508 & 0.393700787401575 \end{pmatrix} = \begin{pmatrix} 0.9997636963856755 \\ 0.14743069476609796 \\ 0.6971436145866375 \\ 0.9600649764431692 \\ 0.8061563923248787 \\ 1.0955499296227986 \\ 1.2632149447990955 \\ 1.9301380869299334 \\ 1.9025657279036454 \\ 0.9353509480767014 \end{pmatrix}$$

Comparando podemos concluir que el individuo más adaptado es el de índice 8 pues la función objetivo obtiene su valor más alto con el conjunto de coordenadas que representa su carga genética.

### 5. SELECCIÓN Y CRUCE

Una vez evaluada la adecuación de cada individuo se seleccionan por sorteo de cuerdo con una ruleta *viciada* ya que esta ruleta favorece en la selección a los más aptos. Conforme se puede observar en la figura 2 los sectores asignados a cada

individuo, son proporcionales a su adecuación. El área de los sectores se calcula de acuerdo con

$$p_i = \frac{f_i}{\sum_{k=1}^{n_I} f_k}$$

donde las  $f_i$  son las adecuaciones de cada individuo. Para elaborar la ruleta se efectúan los siguientes pasos:

- Se efectúan las sumas acumuladas de las  $\sigma f_i$  o sea  $\sigma f_i = \sum_{k=1}^{k=i} f_k$ .
- Se normalizan estas sumas dividiéndolas por la suma total  $\sum_{k=1}^{n_I} f_k$

La gráfica normalizada de  $\sigma f_i$  sirve para el sorteo de apareamiento. Una vez escogidos por sorteo, y por parejas, cruzan sus contenidos genéticos para generar su descendencia.

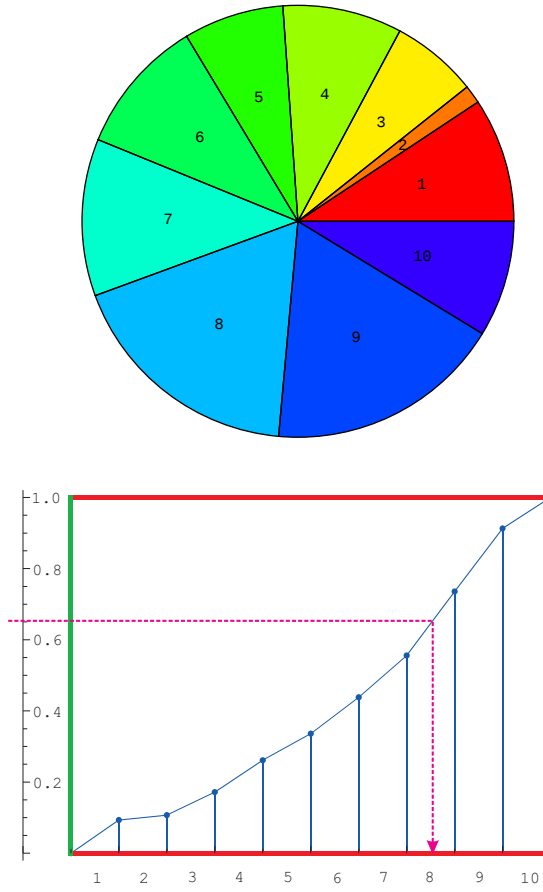


FIGURA 2. Ruleta para sorteo del cruce entre individuos

Una vez seleccionados los individuos, se generará la descendencia a través de la composición de sus cargas genéticas. Así por ejemplo, si se escogen como generadores los individuos de índices 8 y 9, la composición del cromosoma descendiente se puede efectuar por la mezcla de los cromosomas de ambos. Una forma es la mostrada a seguir

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

si los individuos escogidos fueran los de índices 5 y 6 tendríamos

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

en el caso de los individuos 8 y 9 se utilizó una composición de cruce por tres puntos y en el caso de los individuos 5 y 6 se utilizó una composición por dos puntos. De esta forma se completa la nueva población que proseguirá el proceso evolutivo.

## 6. MANTENIENDO LA DIVERSIDAD - MUTACIÓN

El contenido genético de las poblaciones a lo largo del proceso evolutivo queda limitado al aporte inicial de la primera población sorteada al acaso. Para evitar que esto conlleve al estancamiento se cuenta con la acción del medio ambiente induciendo mutaciones con muy baja probabilidad de incidencia. La mutación, altera uno o más componentes del cromosoma, con un elemento del alfabeto, independientemente de su valor actual. A guisa de ejemplo el individuo de índice 2

$$[0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0] \rightarrow [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0]$$

## 7. ALGORITMO

Estas operaciones pueden ser arregladas de manera sistemática en forma de procedimiento general

## 8. VERSIÓN REAL

En la versión real, el contenido individual es formado por un conjunto de números reales asociados a cada una de las coordenadas del espacio de búsqueda. Estas coordenadas son inicialmente generadas aleatoriamente en  $\Omega \subset \mathbb{R}^n$ .

## 9. CRUCE Y MUTACIÓN

Estas dos operaciones se efectúan conjuntamente de acuerdo con el procedimiento:

- Escoge dos progenitores según sorteo a la ruleta
- Se determina el contenido del cromosoma descendiente de acuerdo con la combinación

$$X_d = \alpha X_1 + (\alpha - 1)X_2$$

donde  $X_1$ ,  $X_2$ ,  $X_d$  son los puntos en  $\Omega \subset \mathbb{R}^n$  asociados a cada uno de los progenitores y al descendiente, con  $-0.1 \leq \alpha \leq 1.1$  siendo que  $\alpha$  es generado aleatoriamente en su intervalo.

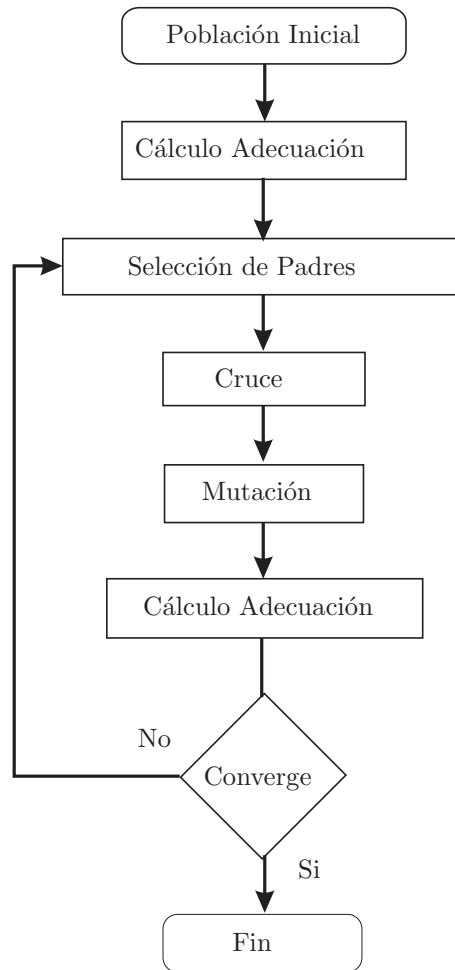


FIGURA 3. Diagrama de Bloques del Procedimiento General

#### 10. CONSIDERACIONES GENERALES

El proceso en detalle envuelve un filtrado aleatorio en cada una de sus etapas.

- El número de puntos de cruce y su ubicación en el caso binario.
- La mutación, a nivel de componente del cromosoma en el caso binario.
- Selección con o sin elitismo. El elitismo consiste en incorporar en poblaciones posteriores una o más copias del individuo más adecuado.

#### 11. EJEMPLO DE APLICACIÓN DE LOS ALGORITMOS GENÉRICOS

Presentaremos el código **MATLAB** para resolver un problema de compensación de una planta a través del diseño de un controlador PID. El objetivo a alcanzar es el de obtener estabilidad en lazo cerrado con mínimo tiempo de asentamiento. La planta en lazo abierto tiene función de transferencia

$$G_p(s) = \frac{s+5}{(s+1)(s^2+2s+2)}$$

```

global nump denp
nump = [1 5];
denp = conv([1, 1],[1, 2, 2]);
h = gcf;
figure(h)
rlocus(nump,denp),title('Polos de Lazo Cerrado')
[kp, kd, ki] = calculaPID;
numPID = [kd, kp, ki];
denPID = [1, 0];
num = conv(nump,numPID);
den = conv(denp,denPID);
polchar = polisum(num,den);
raices = roots(polchar);
real_part = real(raices);
alfa = max(real_part);
imag_part = real(raices*i);
beta = max(imag_part);
h = gcf;
figure(h+1)
rlocus(num,den),title('Polos de Lazo Cerrado')
axis([2*alfa,1,alfa-beta,-alfa+beta])
grid
hold on
plot(raices,'xk','Linewidth',4)
hold off

function [kp, kd, ki] = calculaPID
rand('seed',sum(100*clock));
options = gaoptimset('PopulationSize', 250,...
                    'EliteCount', 1,...
                    'Generations', 150,...
                    'MutationFcn', @mutationadaptfeasible,...
                    'SelectionFcn', @selectionroulette,...
                    'CrossoverFraction',0.7,...
                    'StallTimeLimit', 36000,...
                    'TolFun', 1e-10,...
                    'PlotFcns', @gaplotbestf);
LB = [0; 0; 0];
UB = [10; 10; 10];
[X0,val] = ga(@objetivoPID,3,[],[],[],[],LB,UB,[],options);
kp = X0(1);
kd = X0(2);
ki = X0(3);

```

```
s = sprintf(' kp = %f\n kd = %f\n ki = %f\n val = %f\n',kp,kd,ki,val);
disp('parametros del controlador')
disp(s)
```

```
function f = objetivoPID(X)
global nump denp teta alfa
kp = X(1);
kd = X(2);
ki = X(3);
numPID = [kd, kp, ki];
denPID = [1, 0];
num = conv(nump,numPID);
den = conv(denp,denPID);
polchar = polisum(num,den);
raices = roots(polchar);
real_part = real(raices);
f = max(real_part);
```

```
function ps = polisum(p1,p2)
n1 = length(p1);
n2 = length(p2);
n0 = max(n1,n2);
pa = zeros(1,n0);
pb = zeros(1,n0);
pa(n0-n1+1:n0) = p1(1:n1);
pb(n0-n2+1:n0) = p2(1:n2);
ps = pa + pb;
```

## 12. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS - OEP

(*Extraído de notas de Namrta Khemka, khemka@cpsc.ucalgary.ca*)

En OEP la búsqueda del punto óptimo se realiza a través de una población de individuos  $i$  con vector de localización  $\vec{p}_i$  que representan una solución potencial al problema.

La bondad del proceso de optimización es medida a través de una función de adecuación  $\mathfrak{F}$ . Aplicando  $\mathfrak{F}$  a los vectores solución, se puede evaluar su adecuación relativa. Así, la adecuación del  $i$ -ésimo individuo se define por  $\mathfrak{F}(\vec{p}_i)$ .

La adecuación individual puede también interpretarse como su fuerza de persuasión sobre otros individuos. Cuán más persuasivo sea un individuo, más seguidores tendrá.

Además de su localización  $\vec{p}_i$ , cada individuo posee también una velocidad absoluta  $\vec{v}_i$  que determina en qué dirección y con qué intensidad se moverá el individuo en el espacio de búsqueda a cada intervalo de tiempo.

La evolución del enjambre o población de individuos se desarrolla en pasos discretos, cuando los individuos o potenciales soluciones, vuelan a través del espacio de búsqueda con sus vuelos determinados por las interacciones recíprocas e inerciales.

**12.1. Pseudo Código.** El procedimiento general del sistema OEP sigue:

1. Inicializar el enjambre (población) de partículas asignando localizaciones y velocidades

$$P = (\vec{p}_1, \dots, \vec{p}_i, \dots, \vec{p}_N), \quad V = (\vec{v}_1, \dots, \vec{v}_i, \dots, \vec{v}_N)$$

2. Evaluar la adecuación de todas las partículas del enjambre

$$\mathfrak{F}(P) = (\mathfrak{F}(\vec{p}_1), \dots, \mathfrak{F}(\vec{p}_i), \dots, \mathfrak{F}(\vec{p}_N))$$

3. Guardar constancia de las localizaciones en que cada partícula ha obtenido la mejor adecuación

$$P^{opt} = (\vec{p}_1^{opt}, \dots, \vec{p}_i^{opt}, \dots, \vec{p}_N^{opt})$$

4. Guardar constancia de la posición con la mejor adecuación

$$\vec{p}_g^{opt} = \arg \max_{\vec{p} \in P} (\mathfrak{F}(\vec{p}))$$

5. Actualizar la velocidad de las partículas de acuerdo con

$$\vec{v}_i \leftarrow \vec{v}_i + \psi_1(\vec{p}_i^{opt} - \vec{p}_i) + \psi_2(\vec{p}_g^{opt} - \vec{p}_i), \quad \{i = 1, \dots, N\}$$

6. Actualizar la localización de las partículas con las nuevas velocidades

$$\vec{p}_i \leftarrow \vec{p}_i + \vec{v}_i, \quad \{i = 1, \dots, N\}$$

7. Terminar si alguna condición es alcanzada ( máximo global, número máximo de pasos, etc.)

8. Continuar en el paso 2.

A seguir se puede observar un algoritmo más detallado.

$\vec{p}_i(t)$  : posición del individuo  $i$  al instante  $t$   
 $\vec{p}_{i\text{best}}$  : mejor valor obtenido para el individuo  $i$   
 $\vec{\nu}_i(t)$  : velocidad (incremental) del individuo  $i$   
 $t := 0$

**For**  $i = 1, \dots, N$  (total de individuos)

//Asignar posiciones iniciales aleatorias,  $\vec{p}_i(0)$ , y velocidades,  $\vec{\nu}_i(0)$ , a cada individuo  $i$

```

For  $d = 1, \dots, D(\text{numero de dimensiones})$ 
     $p_{i,d} := \text{Random}[P_{\text{mín},d}, P_{\text{máx},d}]$ 
     $\nu_{i,d} := \text{Random}[V_{\text{mín},d}, V_{\text{máx},d}]$ 
Next  $d$ 
Next  $i$ 

 $t := 1$ 
Loop
    For  $i = 1, \dots, N$  (total de individuos)
        If  $\mathcal{F}(\vec{p}_i(t)) > \mathcal{F}(\vec{p}_i^{\text{best}})$  then do //  $\mathcal{F}(\cdot)$  evaluar las adecuaciones
            //  $\vec{p}_i^{\text{best}}$  mejor valor hasta el momento para el individuo  $i$ 
            For  $d = 1, \dots, D(\text{numero de dimensiones})$ 
                 $p_{i,d}^{\text{best}} := x_{i,d}$ 
            Next  $d$ 
        End do

         $g := i$ 
        For  $k \in \text{vecindad}_g$ 
            If  $\mathcal{F}(\vec{p}_k^{\text{best}}) > \mathcal{F}(\vec{p}_g^{\text{best}})$  then  $g := k$  //  $g$  es el índice del mejor individuo
            // en el grupo de entorno
        Next  $j$ 
        For  $d = 1, \dots, D(\text{numero de dimensiones})$ 

            // Actualizar los incrementos de movimiento
             $v_{i,d}(t) := v_{i,d}(t-1) + \varphi_1(p_{i,d}^{\text{best}} - p_{i,d}(t-1)) + \varphi_2(p_{g,d}^{\text{best}} - p_{i,d}(t-1))$ 
             $v_{i,d} \in [V_{\text{mín},d}, V_{\text{máx},d}]$ 

            // Actualizar la posición
             $p_{i,d}(t) := p_{i,d}(t-1) + v_{i,d}(t)$ 
             $p_{i,d} \in [P_{\text{mín},d}, P_{\text{máx},d}]$ 
        Next  $d$ 
        // Ahora:  $\vec{p}_i(t) := \vec{p}_i(t-1) + \vec{\nu}_i(t)$ 
    Next  $i$ 
     $t := t + 1$ 

```

Hasta que el criterio de parada se cumpla (e.g., sobre un total de  $\Delta$  iteraciones)

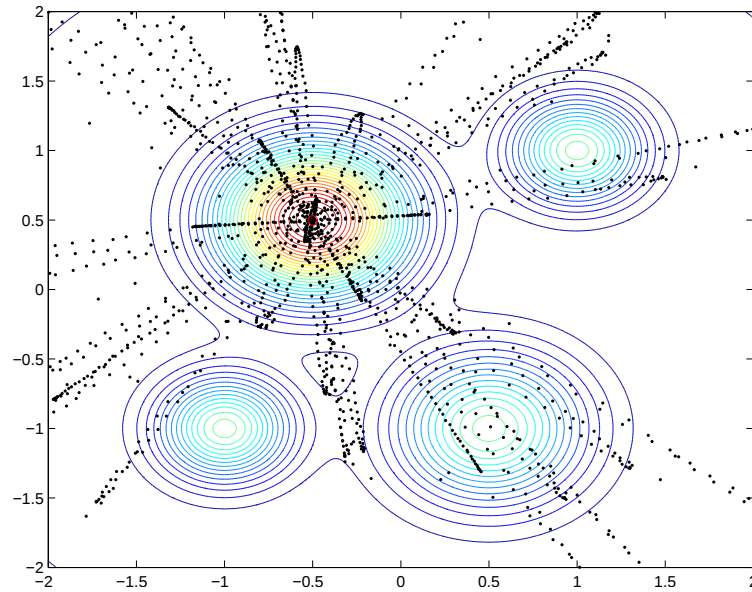


FIGURA 4. Plano de fase del enjambre para la función

$$f = 2e^{-4((x_1+0.5)^2+(x_2-0.5)^2)} + e^{-4((x_1-0.5)^2+(x_2+1)^2)} + e^{-8((x_1+1)^2+(x_2+1)^2)} + e^{-8((x_1-1)^2+(x_2-1)^2)}$$

### 12.2. Código MATLAB.

```
function [Fo,X] = swarm(f,N,M,psi1,psi2,pmin,pmax,vmin,vmax)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [Fo,X] = swarm(f,N,M,psi1,psi2,pmin,pmax,vmin,vmax)
%
%   f = funcion adecuacion
%   N = dimension del enjambre
%   M = numero de epocas
%   psi1 = multiplicador
%   psi2 = multiplicador
%   pmin = limite inferior de la region de exploracion
%   pmax = limite superior de la region de exploracion
%   vmin = limite inferior de la velocidad
%   vmax = limite superior de la velocidad
%   Fo = valor optimo alcanzado
%   X = coordenadas del punto optimo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Condiciones iniciales
%
D = length(pmin);
P = [];
F = [];
V = [];

for i = 1:N
    p = pmin+rand(D,1).*(pmax-pmin);
    P = [P,p];
    v = vmin+rand(D,1).*(vmax-vmin);
    V = [V,v];
    F = [F,f(p)];
end

plot(P(1,:),P(2,:),'.k')
Fb = F;
Pb = P;
```

```

Fo = max(Fb);
Po = Pb;
io = find(Fo == Fb);
io = io(1);
%
% lazo del proceso
%
for i = 1:M
    for k = 1:N
        F(k) = f(P(:,k));
        if F(k) > Fb(k)
            Pb(:,k) = P(:,k);
            Fb(k) = F(k);
        end
        if F(k) > Fo
            io = k;
            Fo = F(k);
        end
    end

    for j = 1:N
        Po(:,j) = Pb(:,io);
    end
    %
    % actualizacion de velocidades y posiciones
    %
    V = V + psi1*(Pb-P) + psi2*(Po-P);
    P = P + V;
    plot(P(1,:),P(2,:),'.k')
end

X = Po(:,io);

```

ETSII- VIGO

*E-mail address:* cesareo@uvigo.es