OPTIMIZACIÓN

Erik Cuevas, Valentín Osuna, Diego Oliva y Margarita Díaz

CAPÍTULO 7
SISTEMAS INMUNES ARTIFICIALES (AIS)

Introducción

- Técnica inspirada en el funcionamiento del sistema inmune de los mamíferos (teoría de selección clonal):

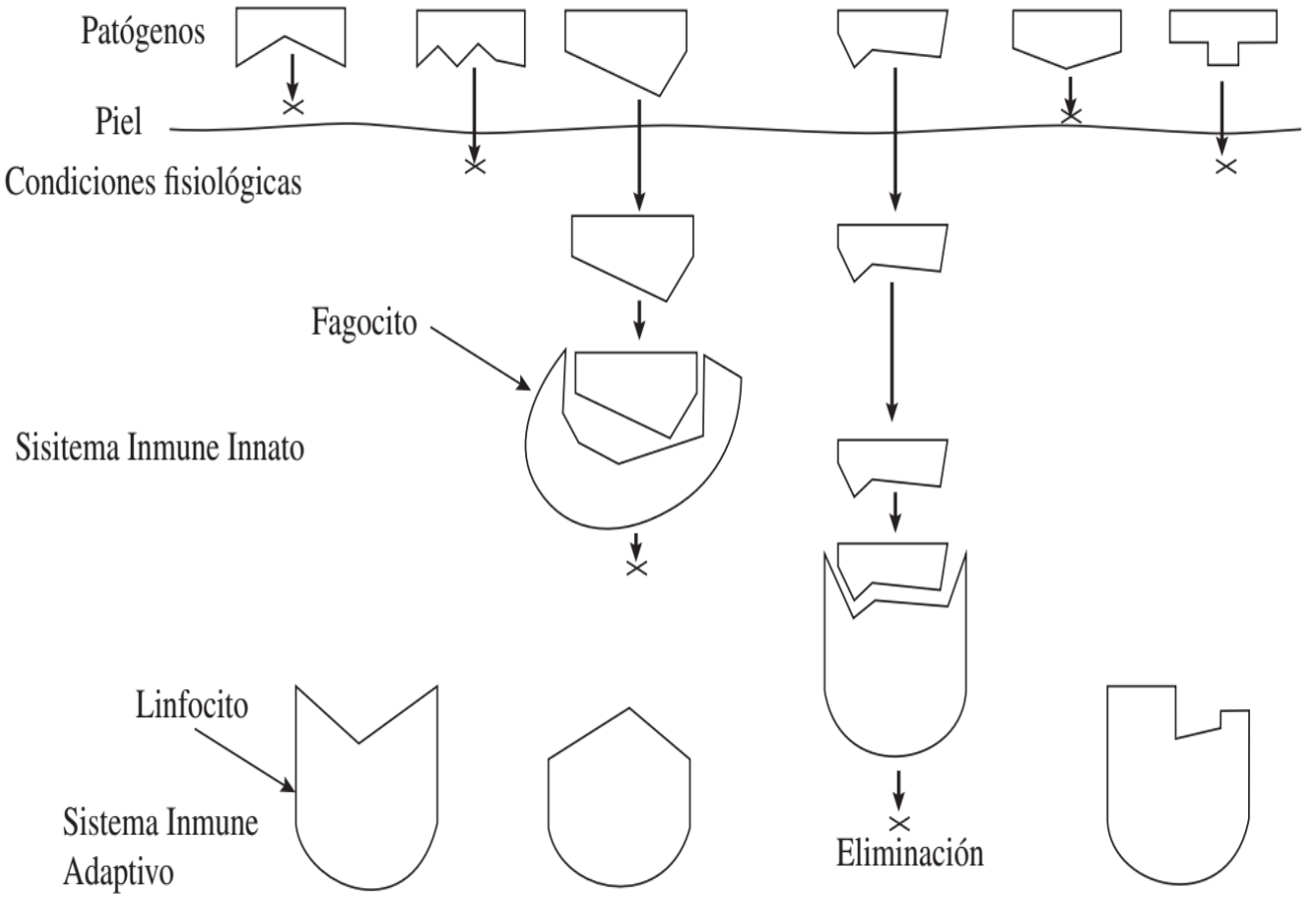
1. Exposición a un conjunto de Ag
2. Producción de Ac
3. Interacciones Ac-Ag

4a. Ac con baja afinidad mueren

4b. Ac con alta afinidad provocan respuesta inmune:

- 4b1. Clonación
- 4b2. Hipermutación
- 4b3. Ataque a Ag

5. Re-selección
6. Memorización



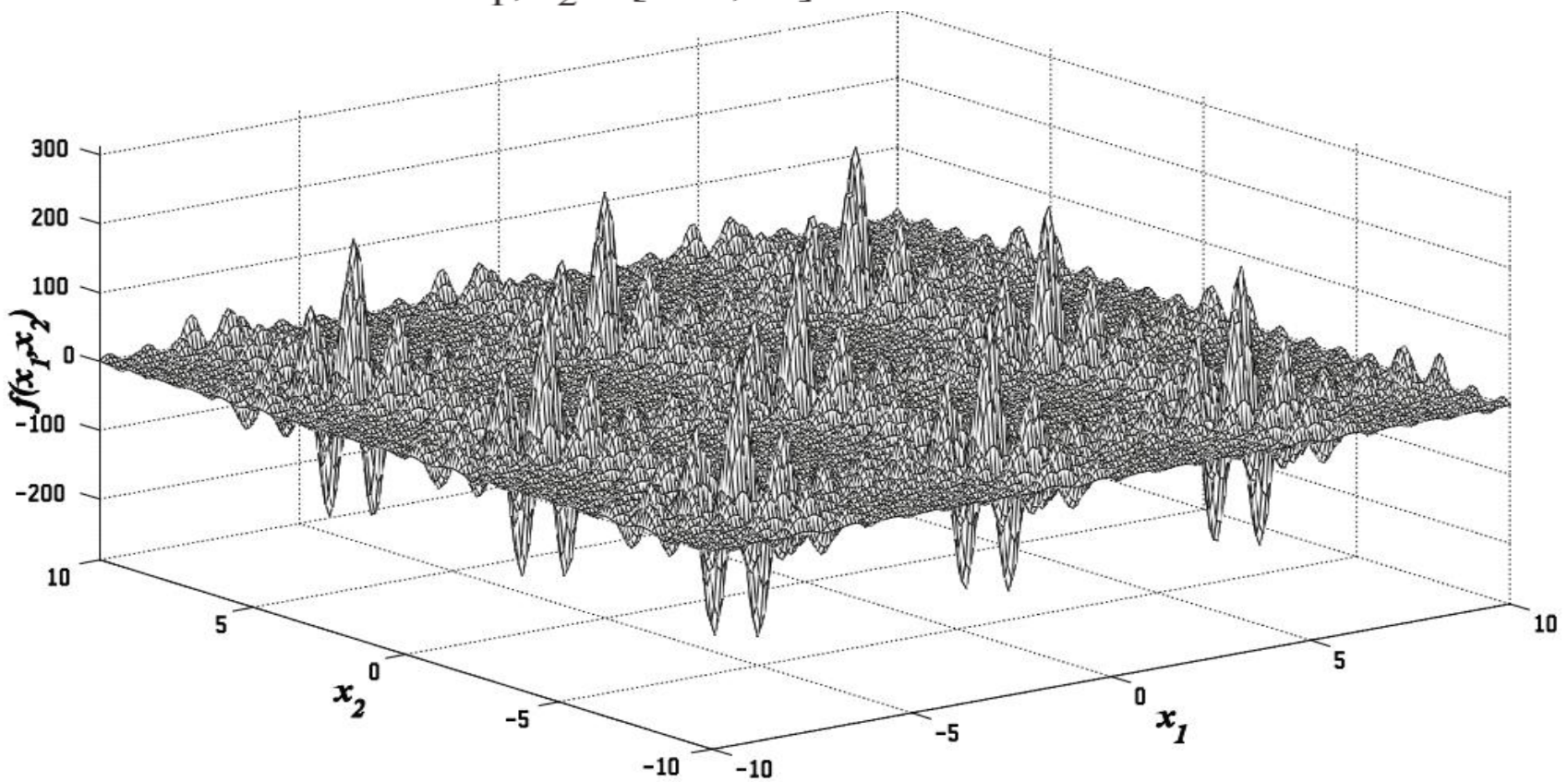
Generalidades

Minimizar

$$f(x) = f(x_1, x_2) = \left(\sum_{i=1}^{5} i \cdot \cos((i+1) \cdot x_1 + i) \right) \left(\sum_{i=1}^{5} i \cdot \cos((i+1) \cdot x_2 + i) \right) \tag{7.1}$$

considerando

$$x_1, x_2 \in [-10, 10]$$



Inicialización

- Los individuos son vectores de números binarios:

$$b_{i,j} = 2 * rand(\cdot) - 1 \qquad (7.2)$$

$$i = 1, \ldots, Np;\ j = 1, \ldots, Nb * d$$

- La población se divide en dos, memoria y remanente:

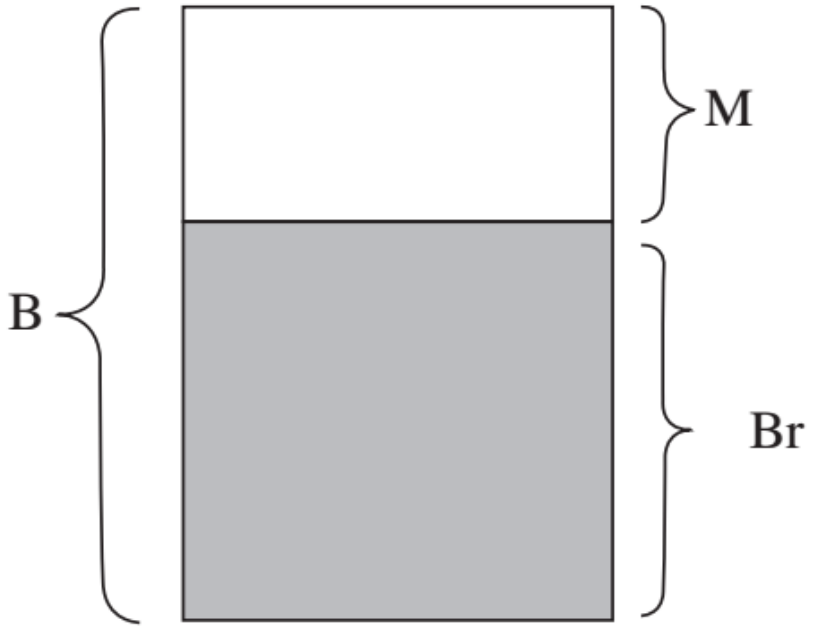


Fig. 7.3. Composición de la población en el ASC.

Clonación

- El operador de reproducción es asexual: cada descendiente se crea a partir de un solo padre, y este operador se aplica solamente a los *n* mejores individuos de la población, cada uno de los cuales serán clonados considerando un cierto porcentaje de toda la población
- Se ordenan los individuos de acuerdo con su correspondiente valor respecto a la función objetivo, se seleccionan los *n* mejores, y por cada uno de ellos se generarán $Np \cdot Pc$ individuos, produciendo una matriz *C* de clones

Hipermutación

- Este operador del algoritmo considera una probabilidad de mutación (pm) para cada uno de los individuos de la matriz binaria de clones *C*:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mutación de los clones en Algoritmo de Selección Clonal
% Erik Cuevas, Valentín Osuna, Diego Oliva y Margarita Díaz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cm=C;
for i1=1:nc
   for i2=1:Nb*d
      if rand()<=pm
         Cm(i1,i2)=~C(i1,i2);
      end
   end
end
Cm(pcs,:) = B(ind(end-n+1:end),:);
```

Programa 7.5. Obtención de la matriz de clones mutados.

Reselección

- Partiendo de una matriz binaria de clones mutados *Cm* que se convierte a una matriz de números reales *Xm*, se evalúa a cada uno de los individuos para generar un vector de valores de *fitness fm*.
- Por cada uno de los *n* individuos a clonar de la población remanente original tenemos *Np* Pc* clones: los mejores individuos de cada uno de esos bloques de clones sobrevivirán a la siguiente generación.

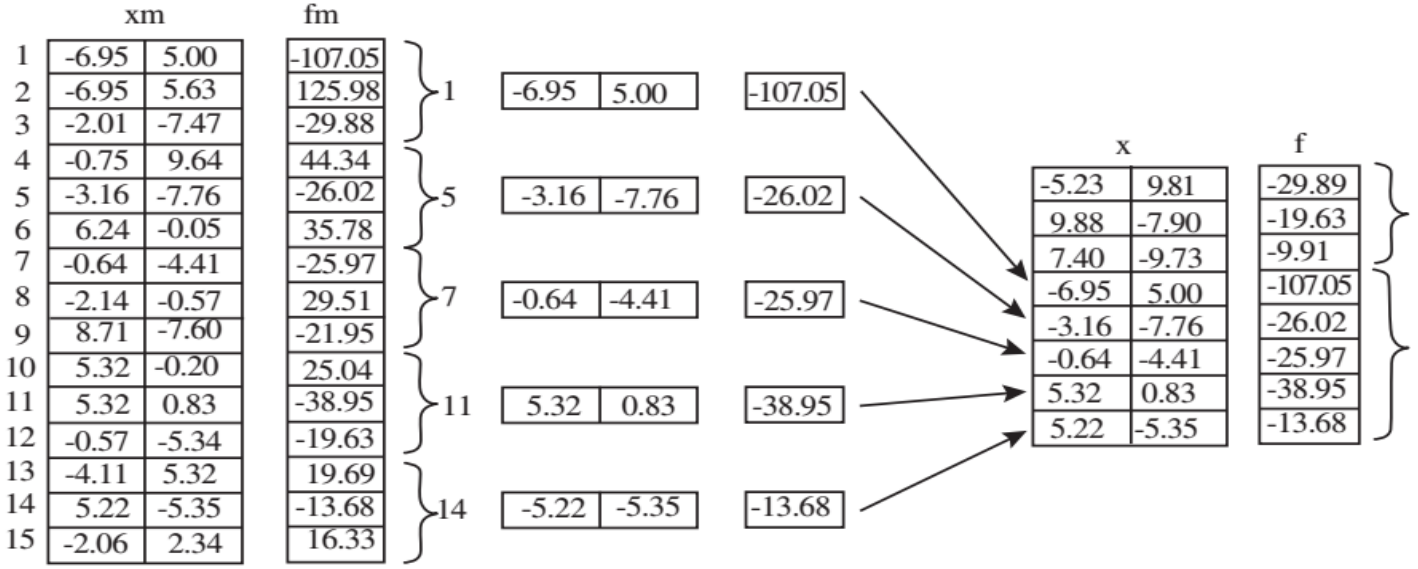


Figura 7.4. Ejemplo de re-selección en el ASC.

Introducción de diversidad

- Con la intención de mantener en todo momento una exploración suficiente de todo el espacio de búsqueda, en el ASC se introducen nuevos elementos que se generan de manera aleatoria
- Este operador se aplica directamente sobre los peores individuos de la población

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Introducción de diversidad en Algoritmo de Selección Clonal
% Erik Cuevas, Valentín Osuna, Diego Oliva y Margarita Díaz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nd = round(Pd*Np);
B(ind(end-nd+1:end),:) = 2 .* rand(nd, Nb*d) - 1;
B(ind(end-nd+1:end),:) = hardlim(B(ind(end-nd+1:end),:));
```

Programa 7.8. Introducción de diversidad en los peores individuos de la población.

Pseudocódigo

Algoritmo 7.1 Algoritmo de Selección Clonal	
1.	Configurar parámetros del algoritmo
2.	Inicializar y evaluar población inicial
3.	Mientras (no se cumpla criterio)
4.	Clonar *n* padres y generar población *C*
5.	Mutar a *C* y generar *Cm*
6.	Re-seleccionar de *Cm* a *n* padres y reemplazarlos en *Br*
7.	Introducir *nd* individuos aleatorios en *Br*(diversidad)
8.	Mostrar resultado