

Metaheurísticas

Unidad 3
Metaheurísticas basadas en Poblaciones

Tema 4 : Evolución Diferencial

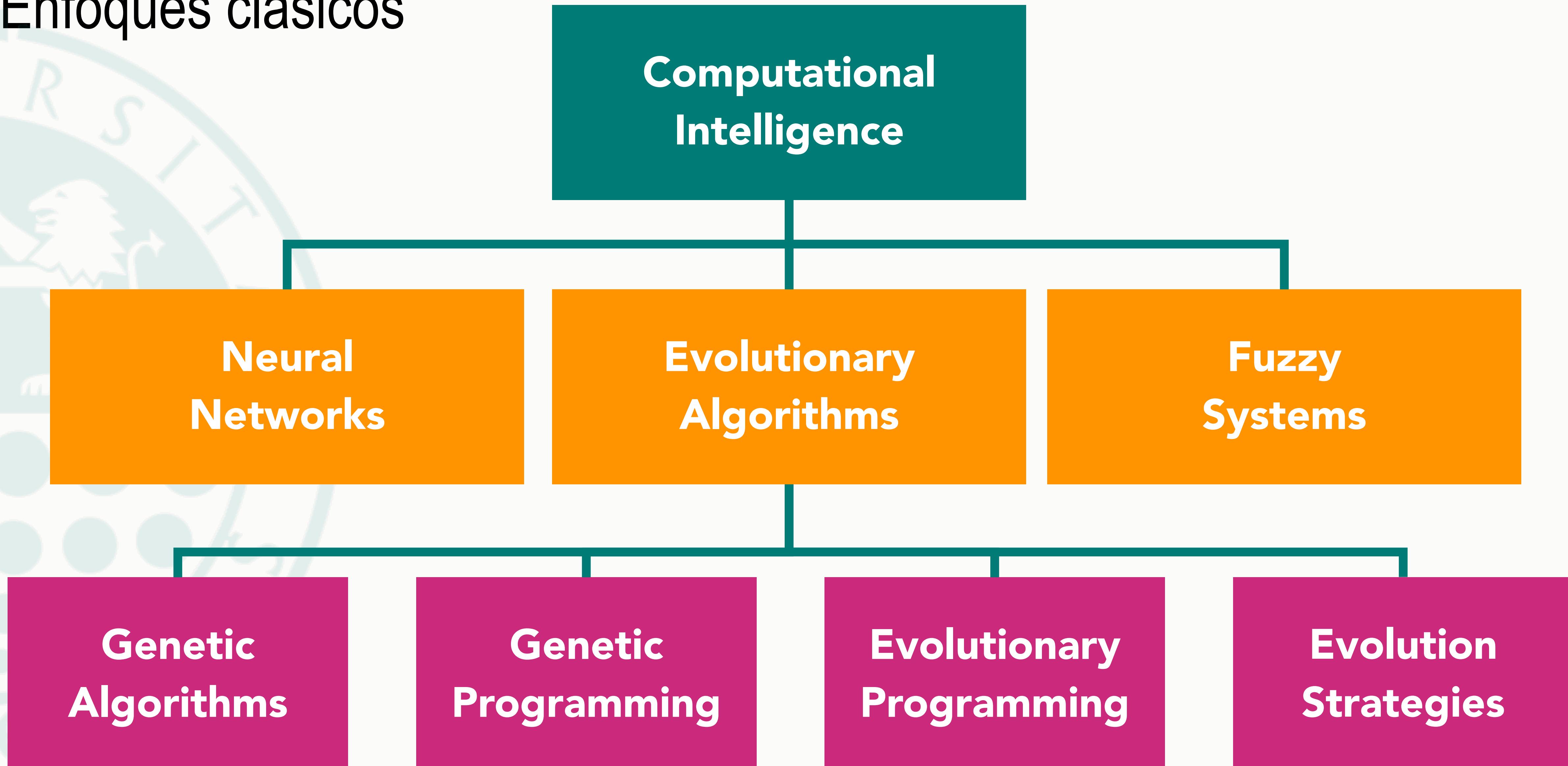
Objetivos

- Presentar una metaheurística basada en poblaciones diferentes de los enfoques clásicos
- Analizar un nuevo enfoque evolutivo para problemas de optimización
- Conocer posibles problemas en los que aplicar este tipo de Metaheurística

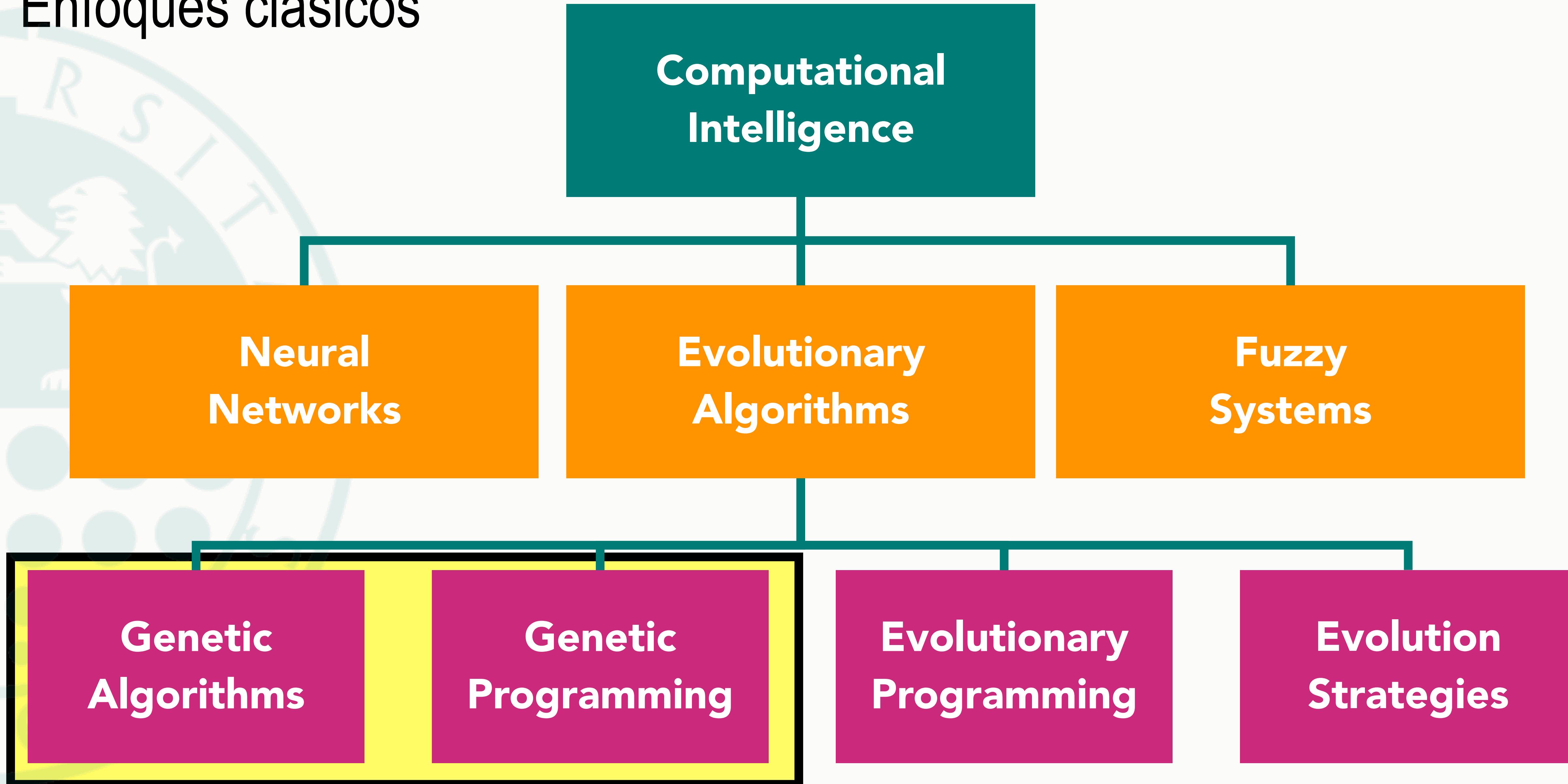
Bibliografía

- [Sto97] R. Storn. Differential Evolution, A simple and efficient heuristic strategy for global optimization over continuous spaces. *Journal of Global Optimization*, 11 (1997) 341–359.
- [Ken05] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)* Springer-Verlag, 2005.

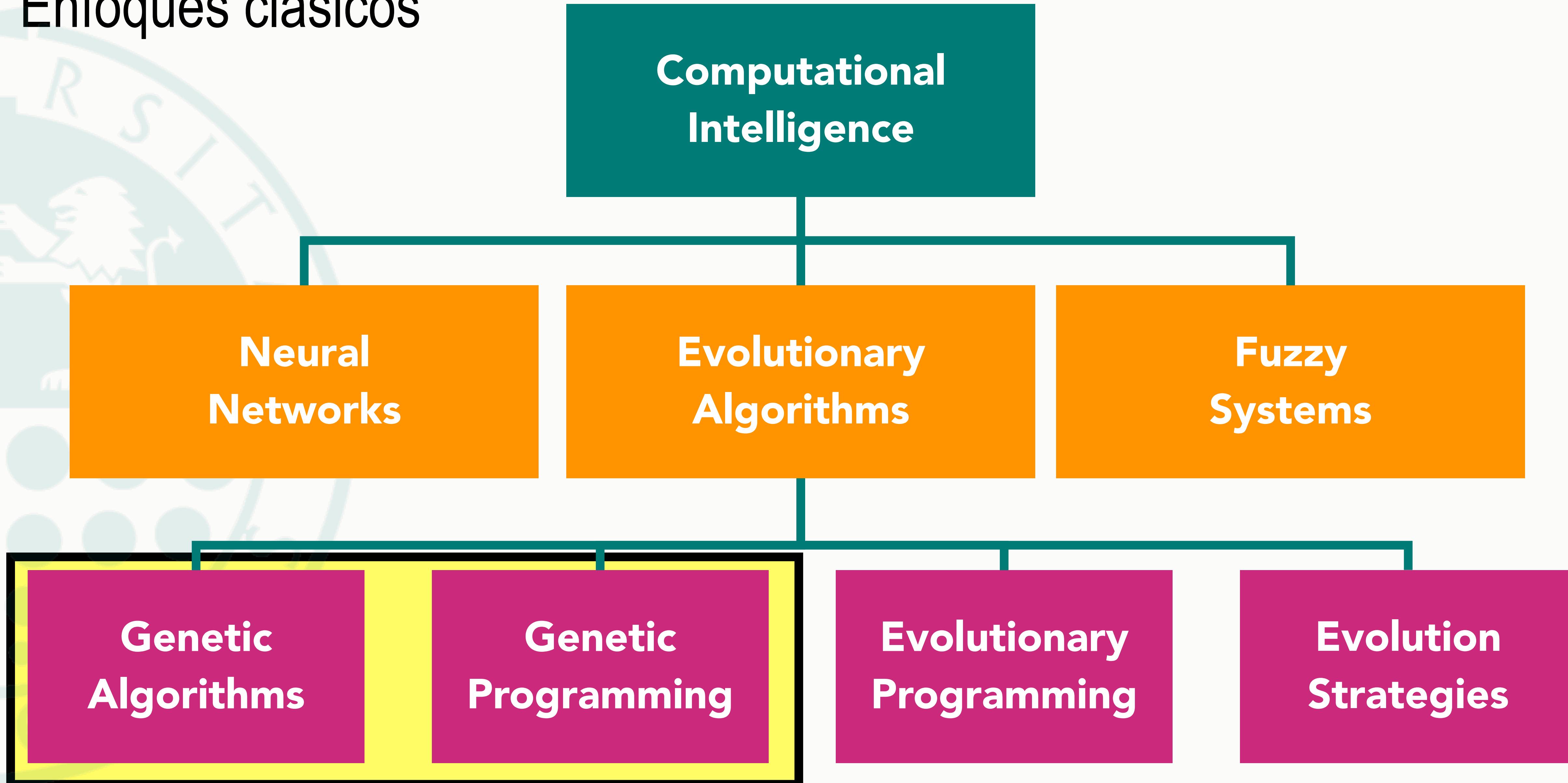
Enfoques clásicos



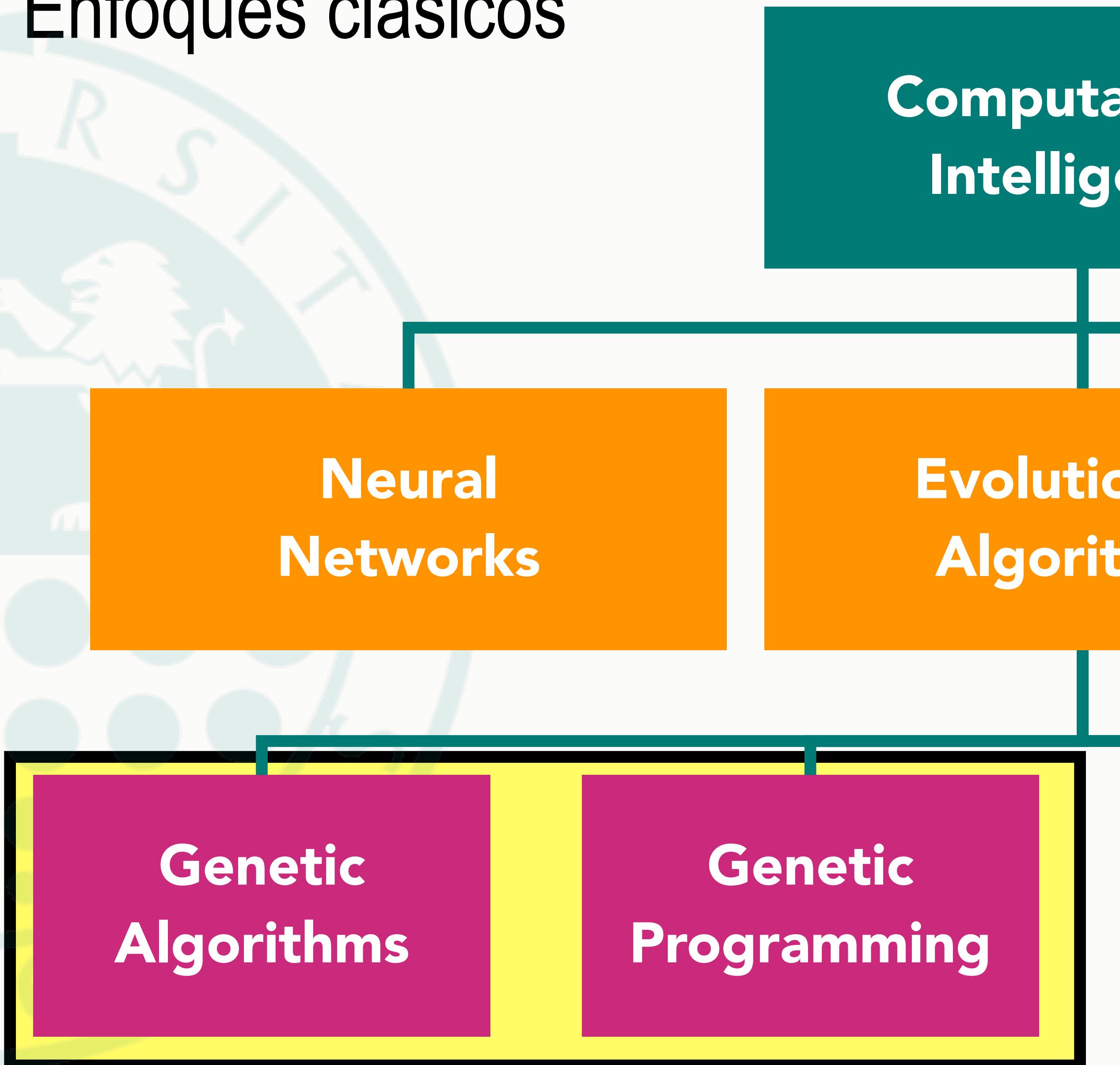
Enfoques clásicos



Enfoques clásicos



Enfoques clásicos



Otro modelo
de CI de
evolución de
poblaciones

Índice

1. Evolución diferencial
2. Variantes de la evolución diferencial
3. Algoritmos avanzados

Evolución diferencial

Modelo evolutivo que enfatiza la mutación, utiliza un operador de cruce/recombinación a posteriori de la mutación

Inicialmente fue propuesto para optimización con parámetros reales

El principal concepto se denomina **OPERADOR DE RECOMBINACIÓN TERNARIO**

Evolución diferencial

El modelo evolutivo clásico tiene el siguiente funcionamiento:

1. Cada descendiente compite con su padre directo
2. El descendiente lo reemplaza Si y solo Si su adaptación es mejor que la de su padre
3. Evolución similar en lo que resta a otros modelos evolutivos

Evolución diferencial

inicialización

- Tenemos una población de n dimensiones
- Cada individuo representa en realidad a un vector

$$x_{i,0} = [x_1{}_{i,0}, x_2{}_{i,0}, \dots, x_n{}_{i,0}]$$

- Cada valor de este vector se genera de forma aleatoria condicionado a unos límites

$$[b_{1,L}, \dots, b_{n,L}] \quad y \quad [b_{1,U}, \dots, b_{n,U}]$$

Evolución diferencial

operador de recombinación ternario

- Este operador se aplica sobre tres individuos:
 - Padre
 - Aleatorio 1
 - Aleatorio 2

$$\text{recombinacion}(P, a_1, a_2) = P + F(a_1 - a_2)$$

Evolución diferencial

operador de recombinación ternario

- Al comienzo todos los individuos se distribuyen por el espacio de forma aleatoria
- La diferencia entre a_1 y a_2 es bastante grande, es decir, implica que el espacio de búsqueda sea lo suficientemente grande => **EXPLORACIÓN**
- Cuando la población converge las soluciones candidatas se van posicionando cerca las una de las otras

Evolución diferencial

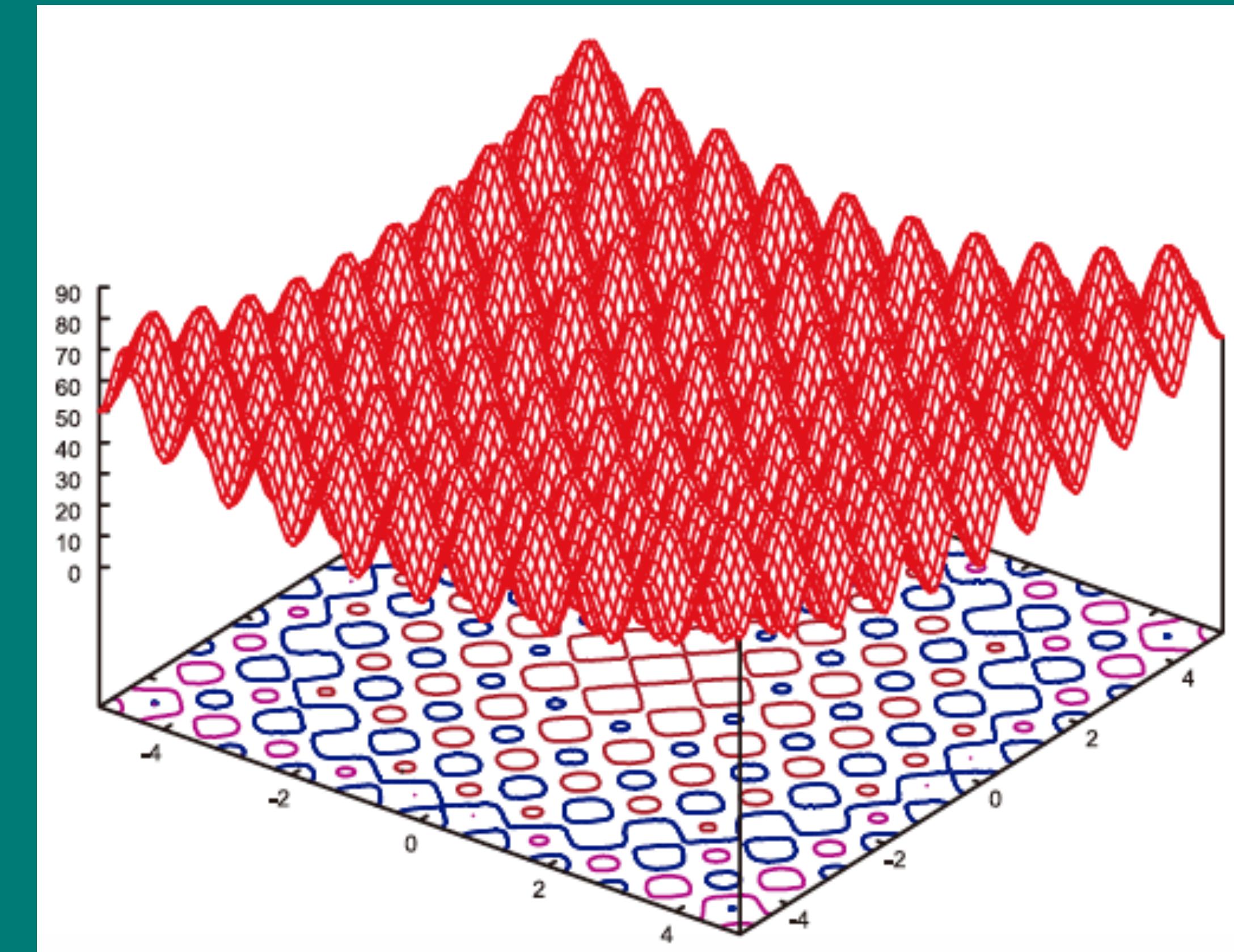
operador de recombinación ternario

- Las diferencias en las distancias por tanto también se hacen más pequeñas
- Este sería un mecanismo simple de auto-adaptación sin necesidad de parámetros adicionales u operaciones
- Conseguimos **EXPLORACIÓN** al comienzo y **EXPLOTACIÓN** al final de forma automática

Ejemplo

Minimizar $f(\vec{x}) = 20 + \sum_{i=1}^n (x_i^2 - 10 \cos(2 * \pi * x_i))$

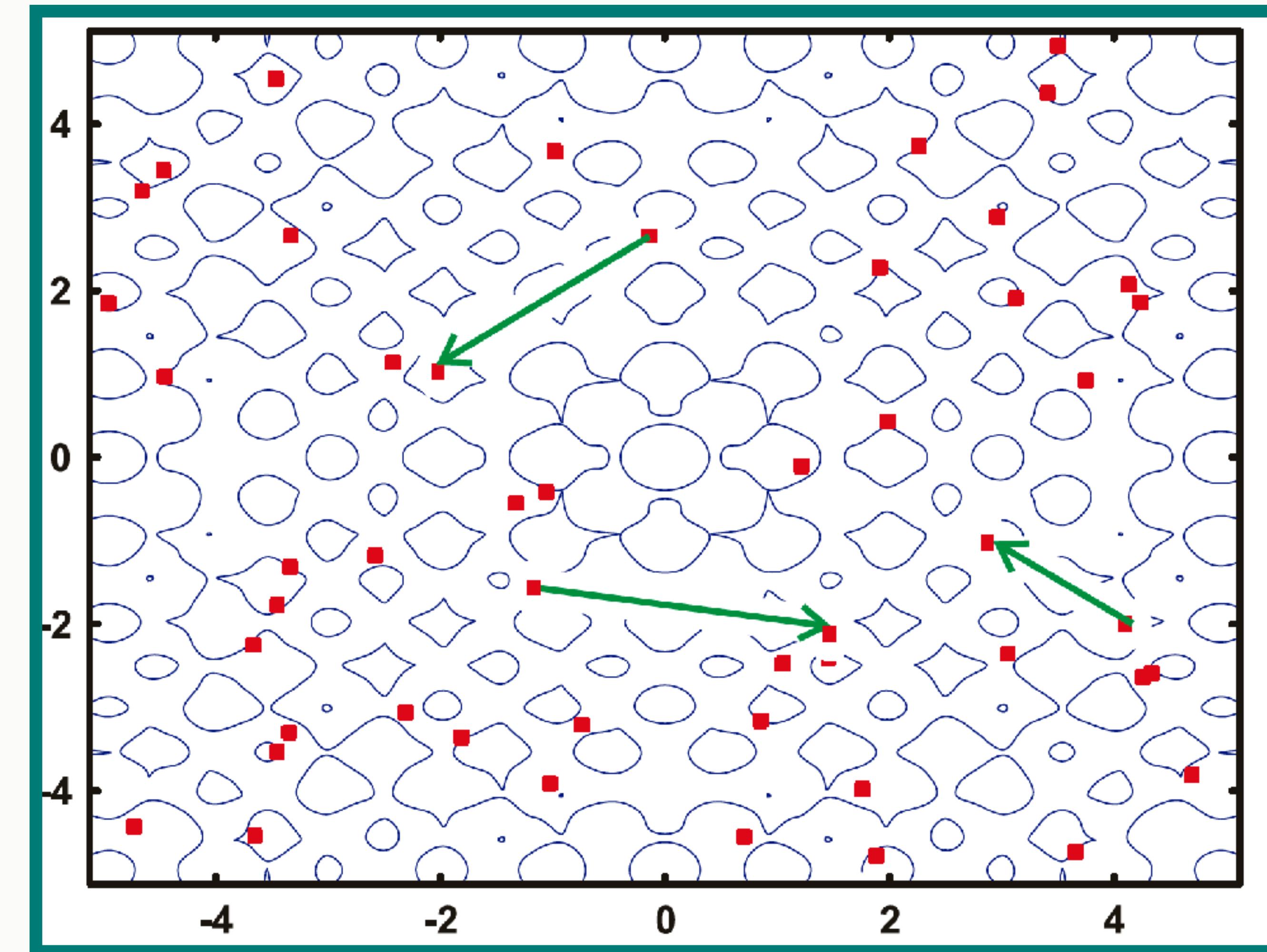
$x \in [-5.12, 5.12]$



Evolución diferencial

ejemplo: minimización de función coseno

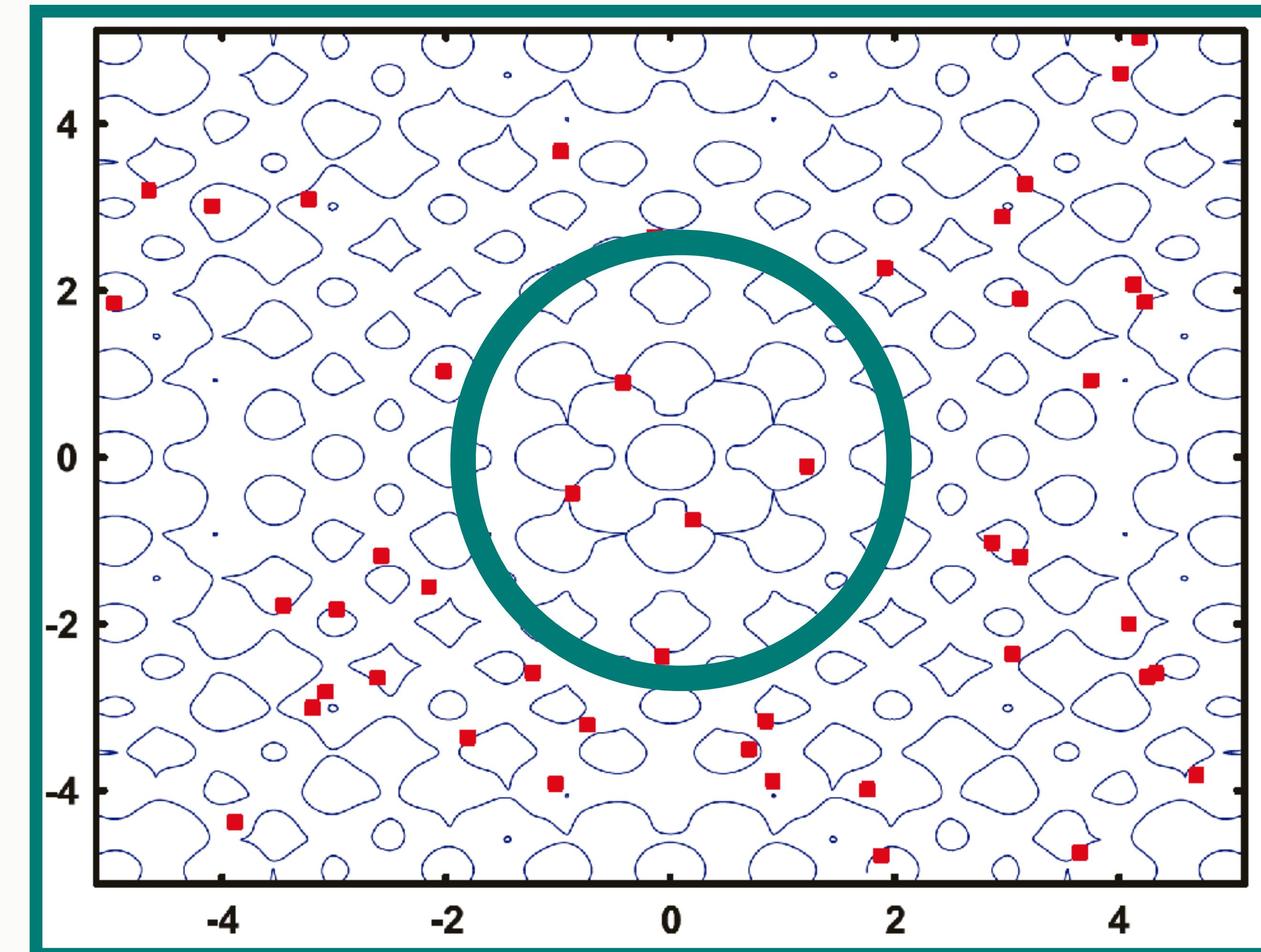
Generación
0



Evolución diferencial

ejemplo: minimización de función coseno

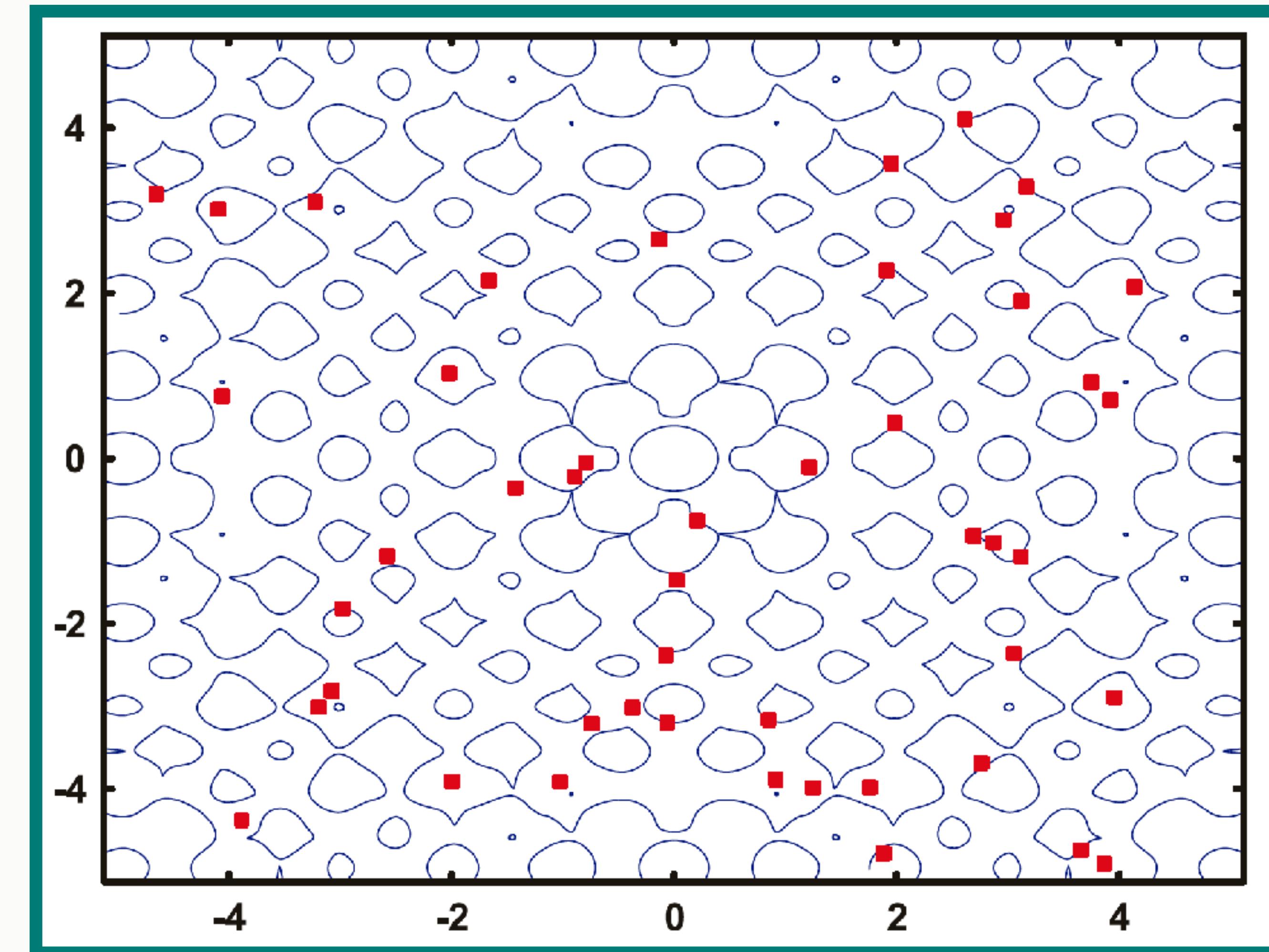
Generación
1



Evolución diferencial

ejemplo: minimización de función coseno

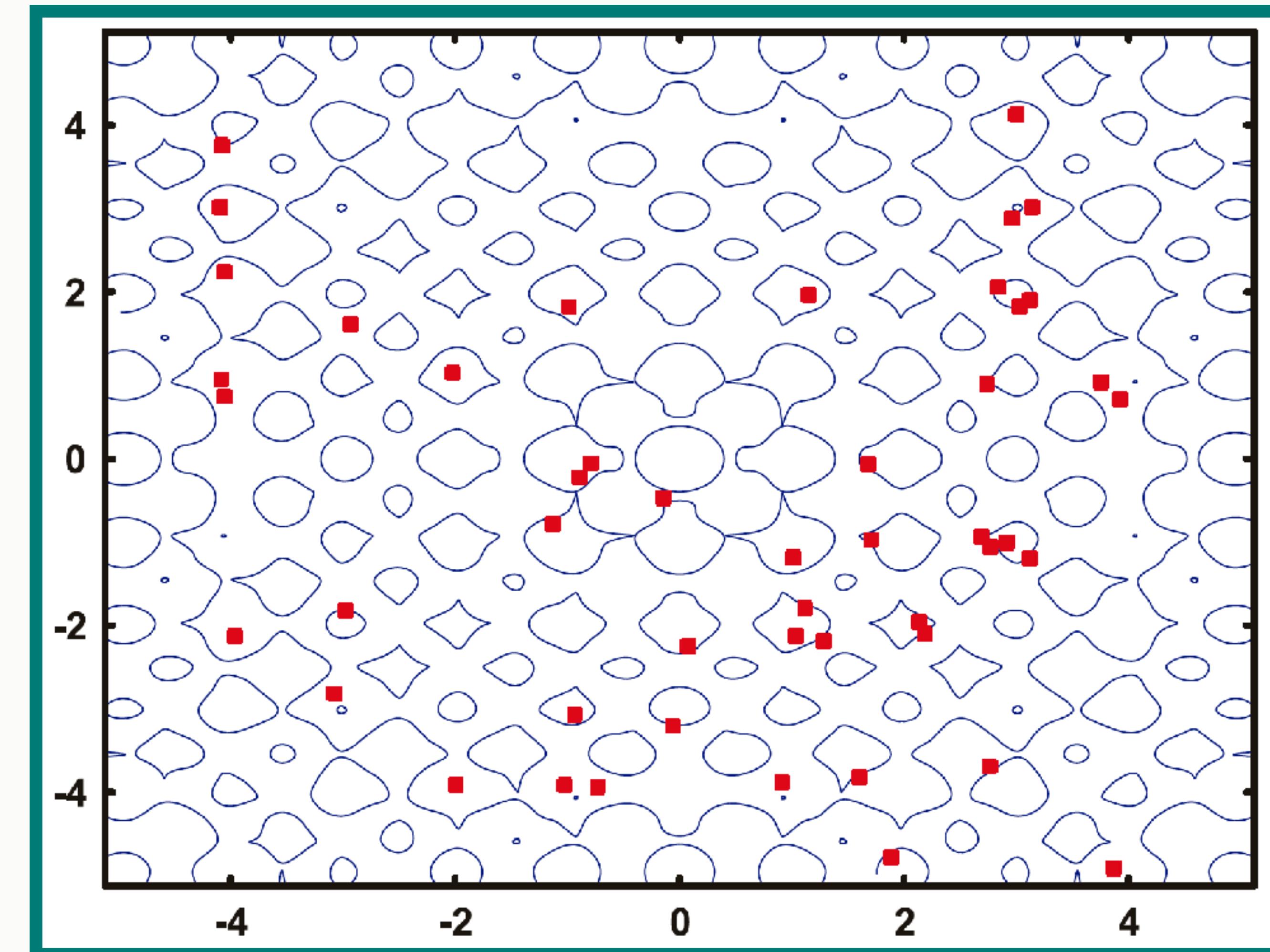
Generación
2



Evolución diferencial

ejemplo: minimización de función coseno

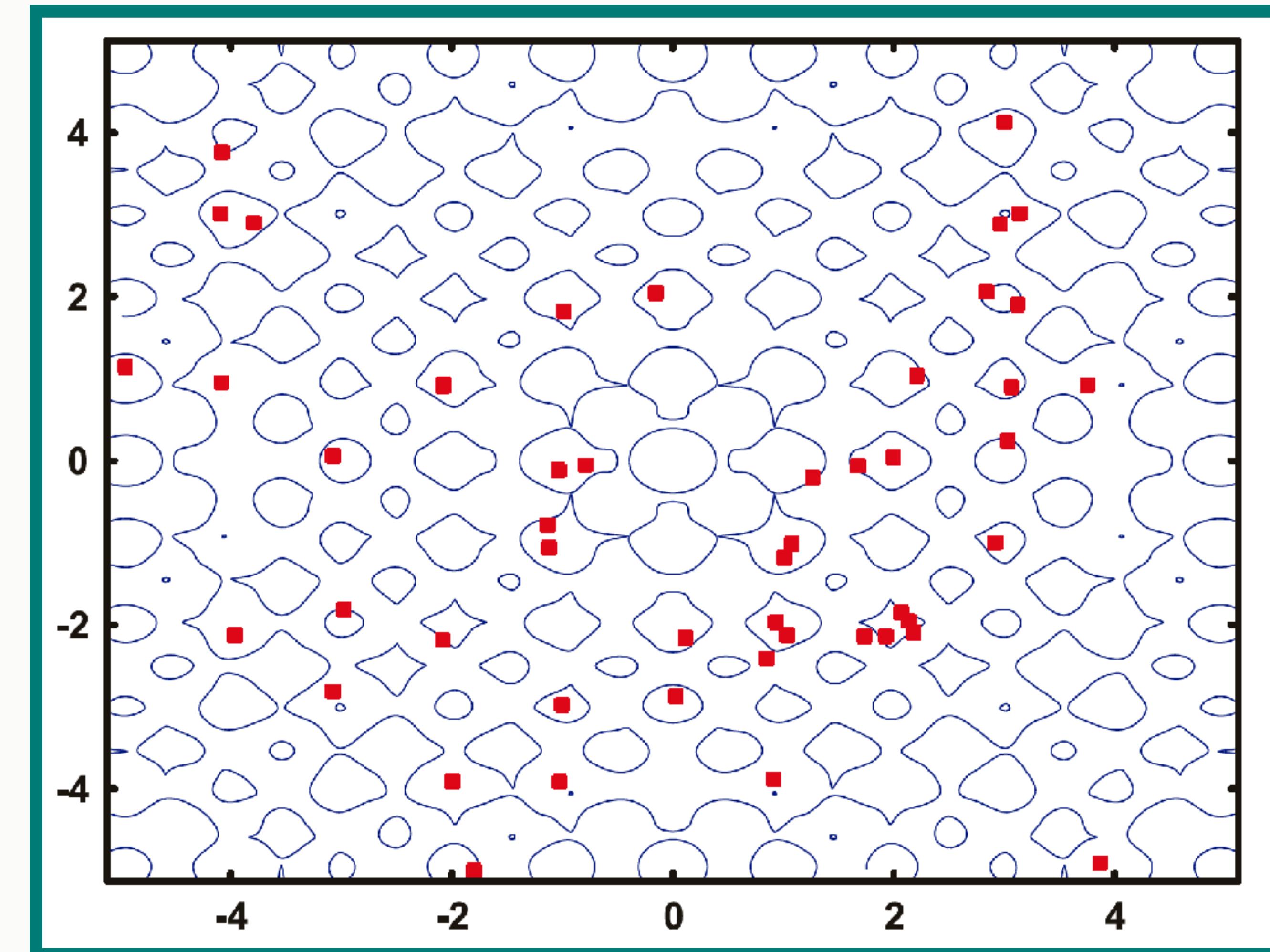
Generación
5



Evolución diferencial

ejemplo: minimización de función coseno

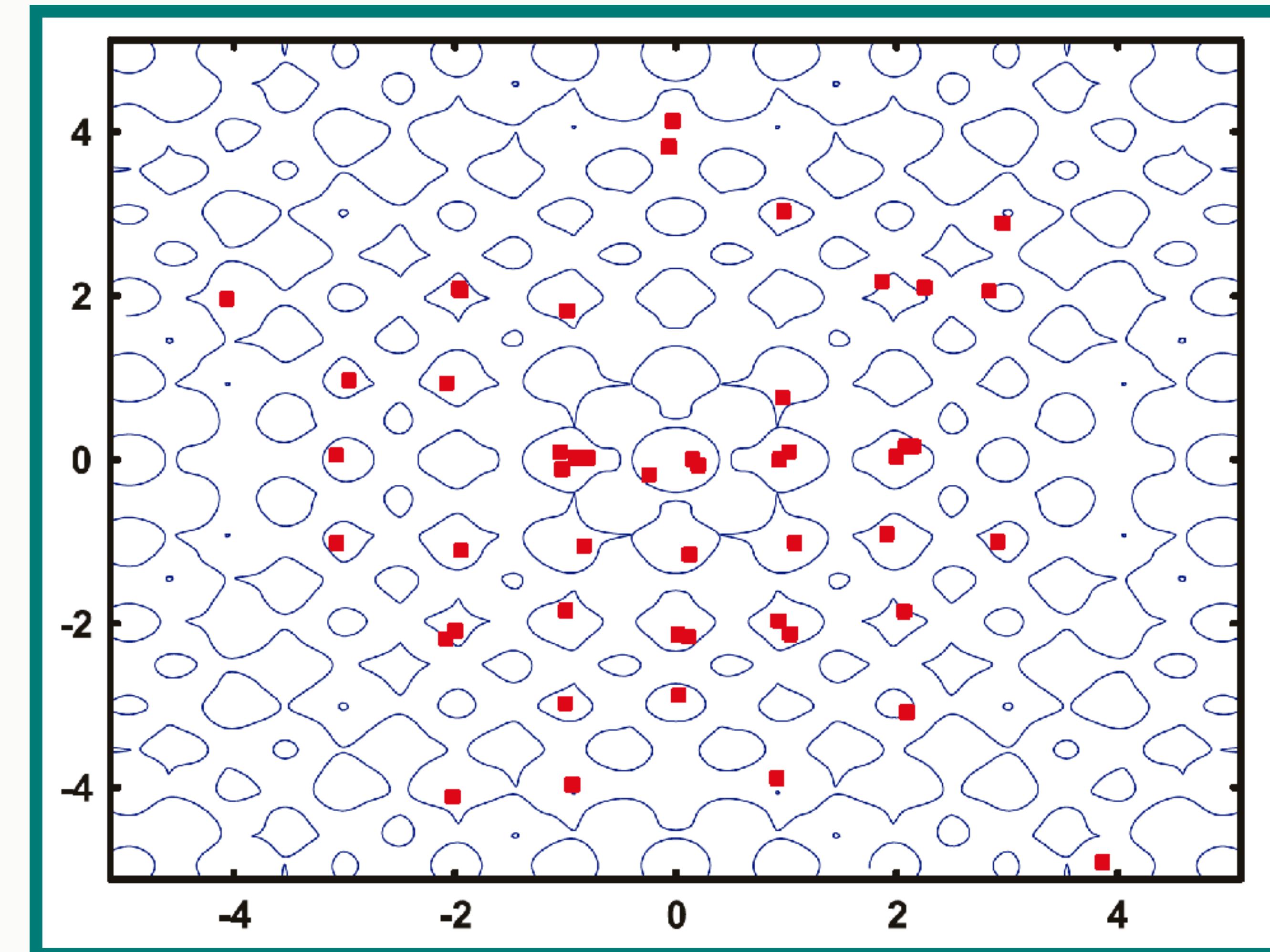
Generación
10



Evolución diferencial

ejemplo: minimización de función coseno

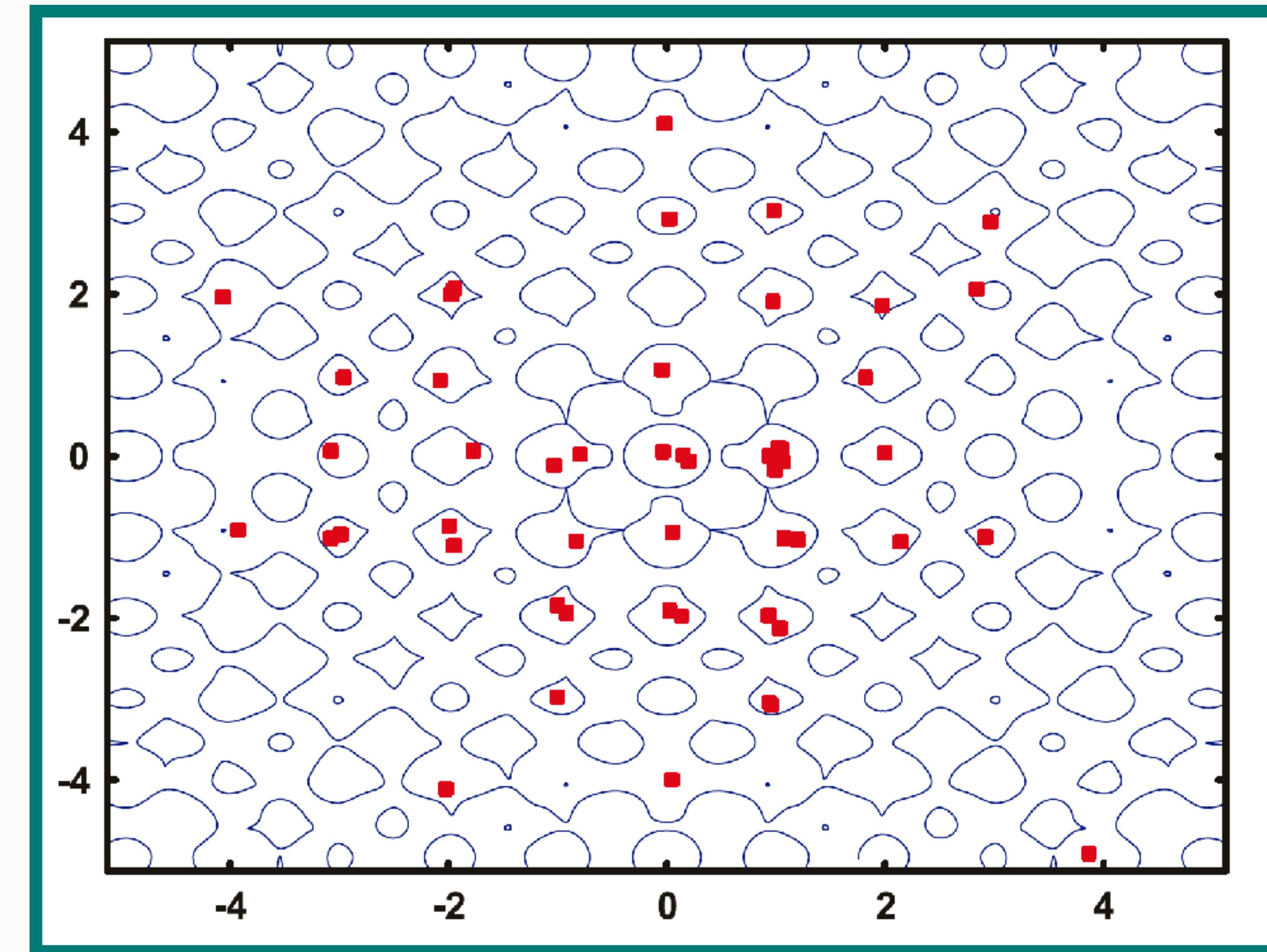
Generación
20



Evolución diferencial

ejemplo: minimización de función coseno

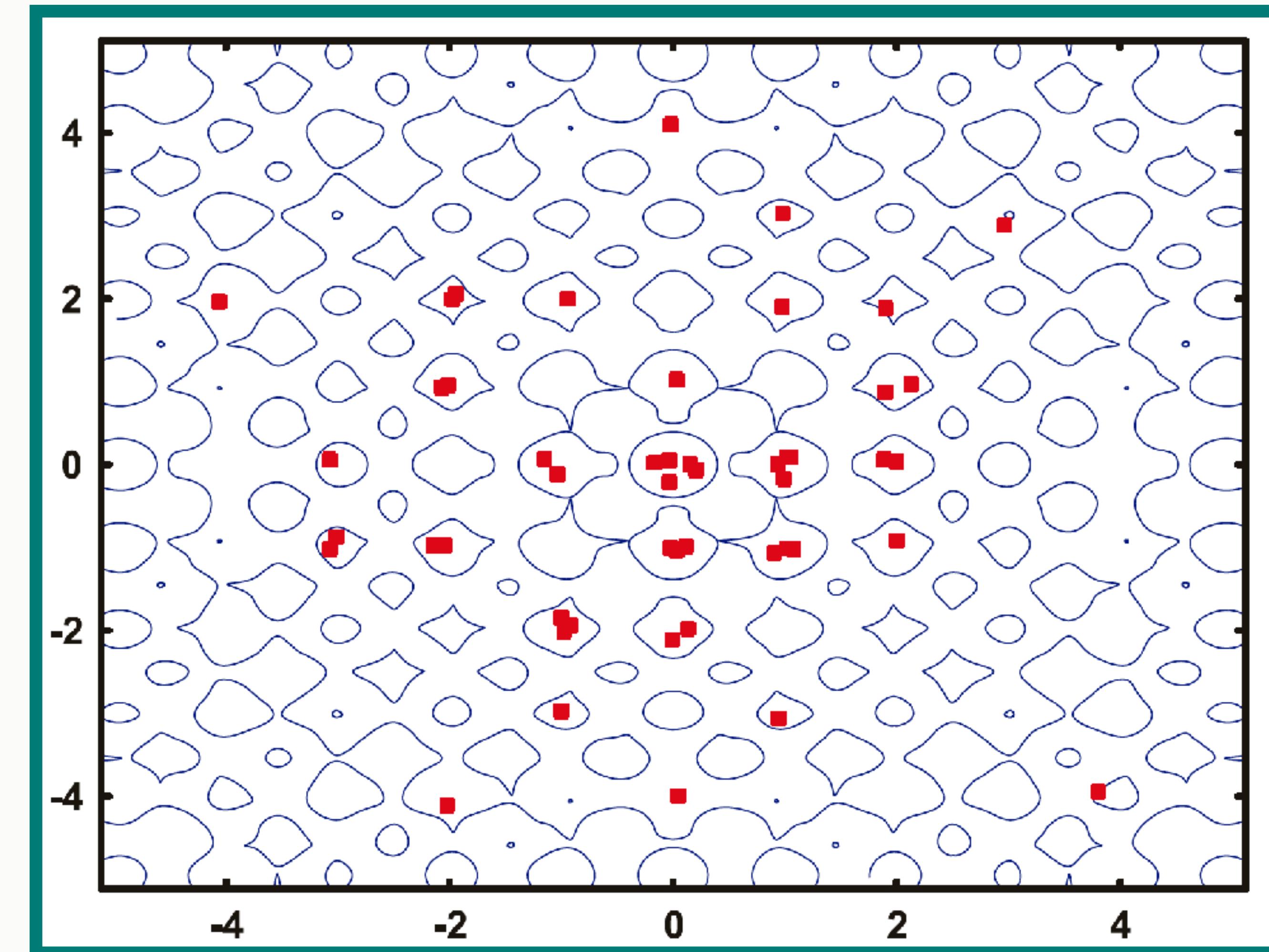
Generación
30



Evolución diferencial

ejemplo: minimización de función coseno

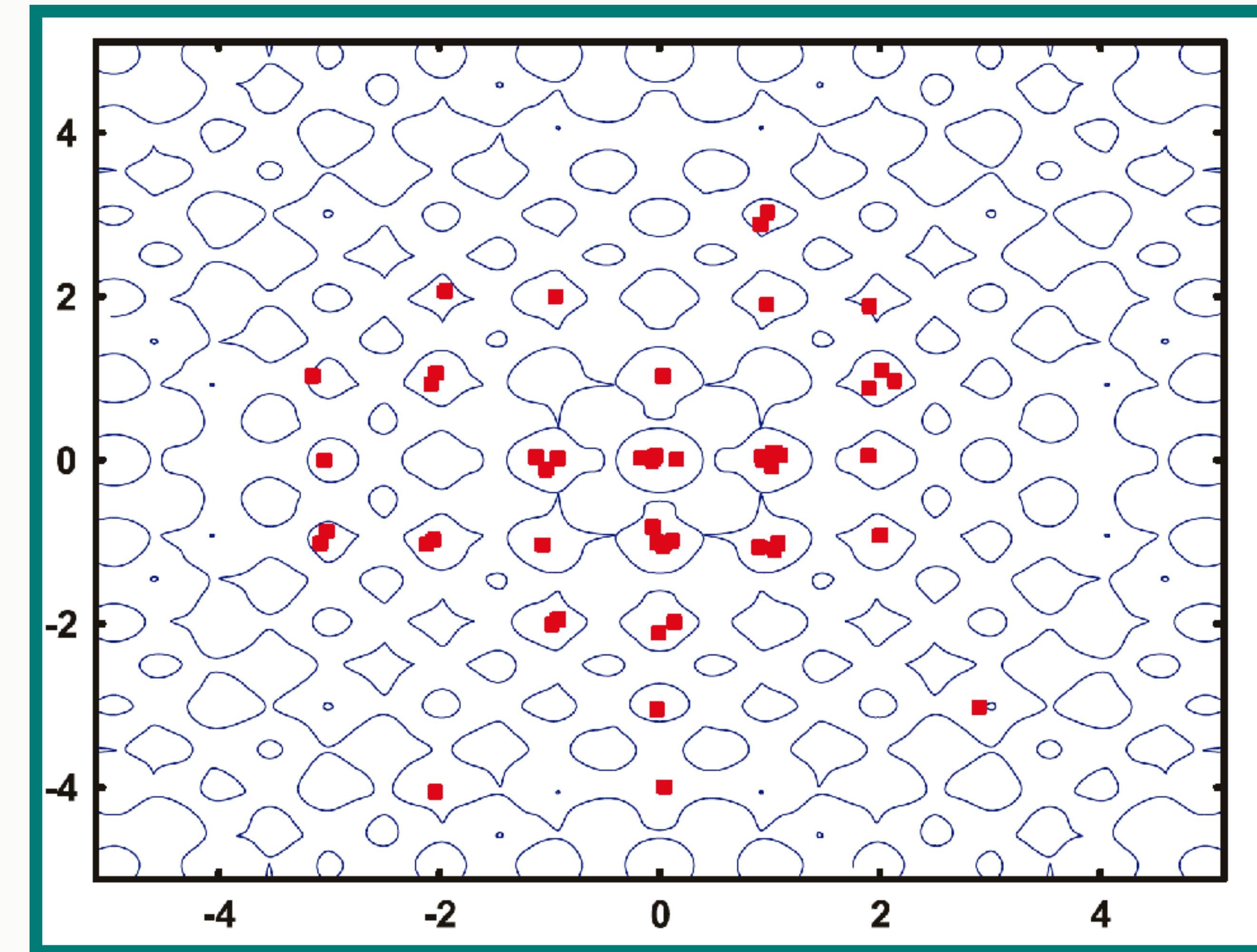
Generación
40



Evolución diferencial

ejemplo: minimización de función coseno

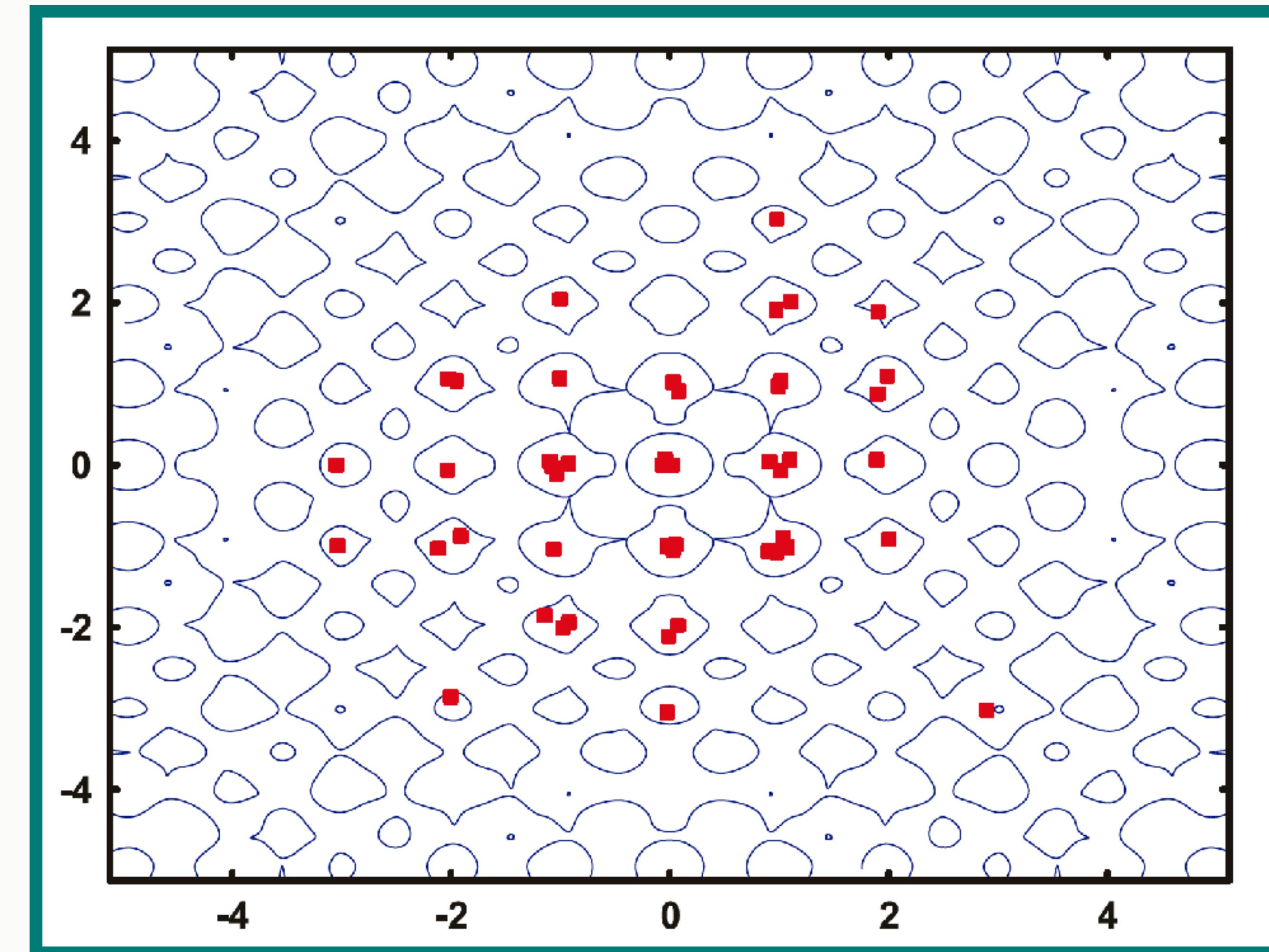
Generación
50



Evolución diferencial

ejemplo: minimización de función coseno

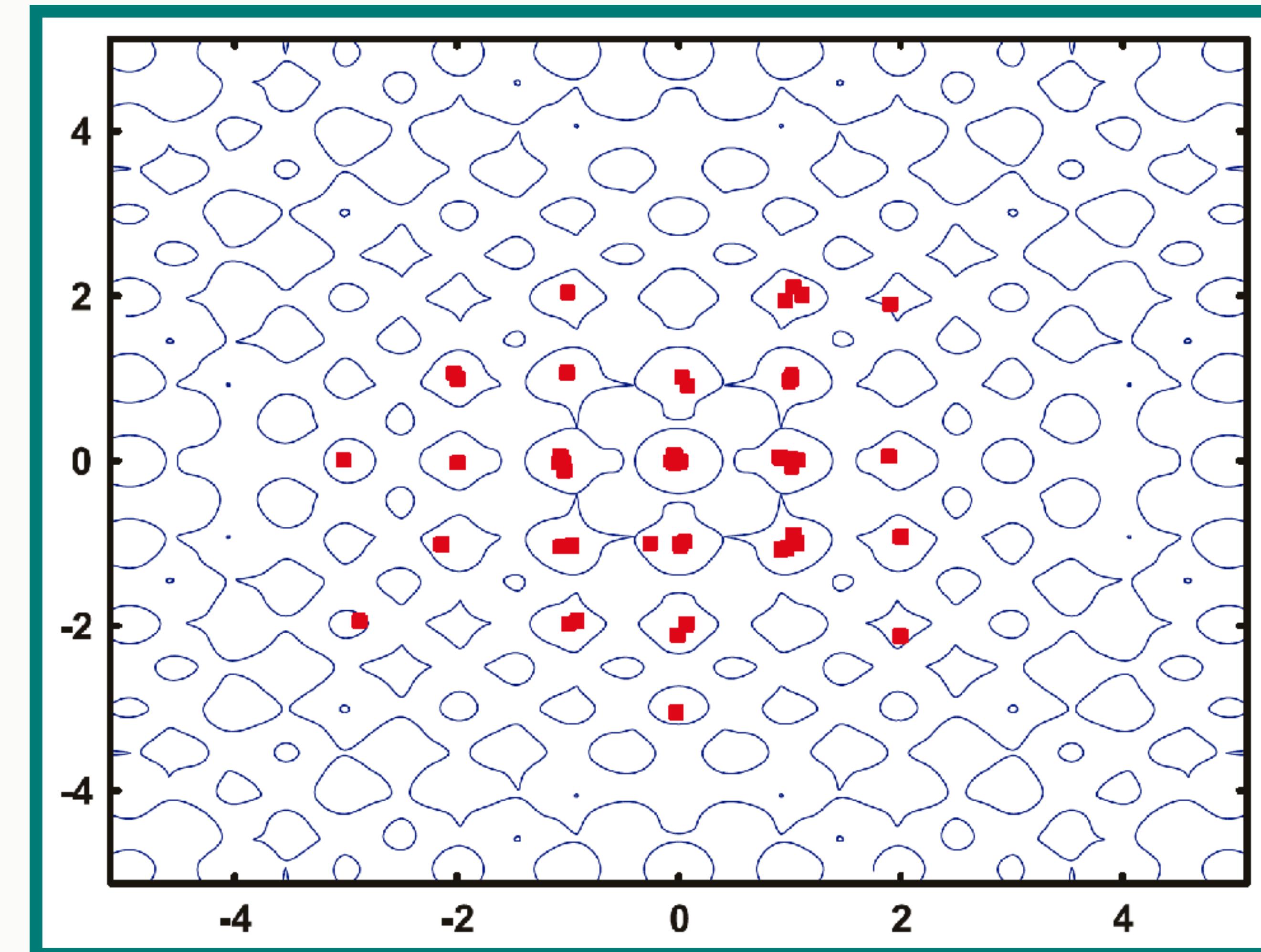
Generación
75



Evolución diferencial

ejemplo: minimización de función coseno

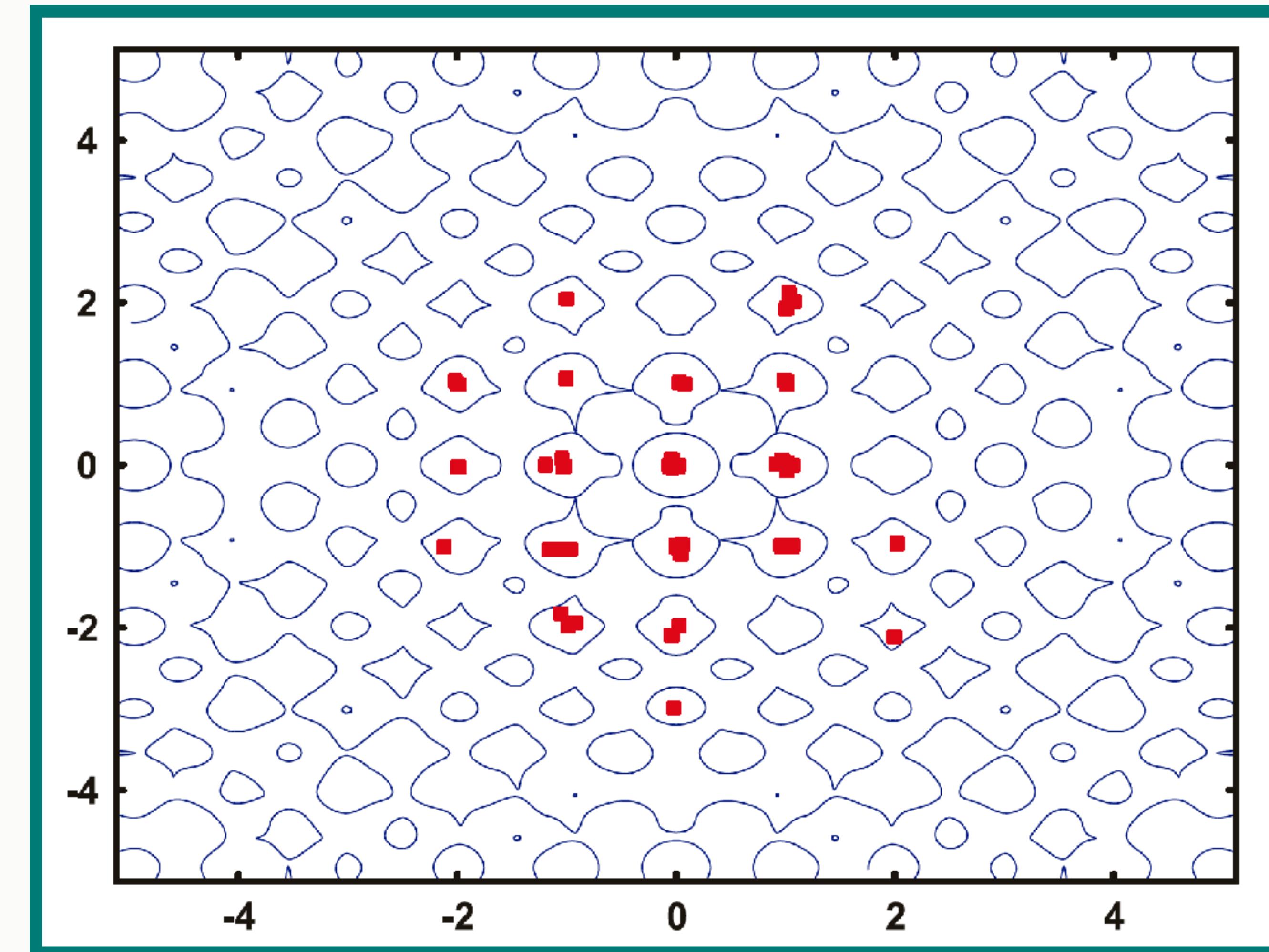
Generación
100



Evolución diferencial

ejemplo: minimización de función coseno

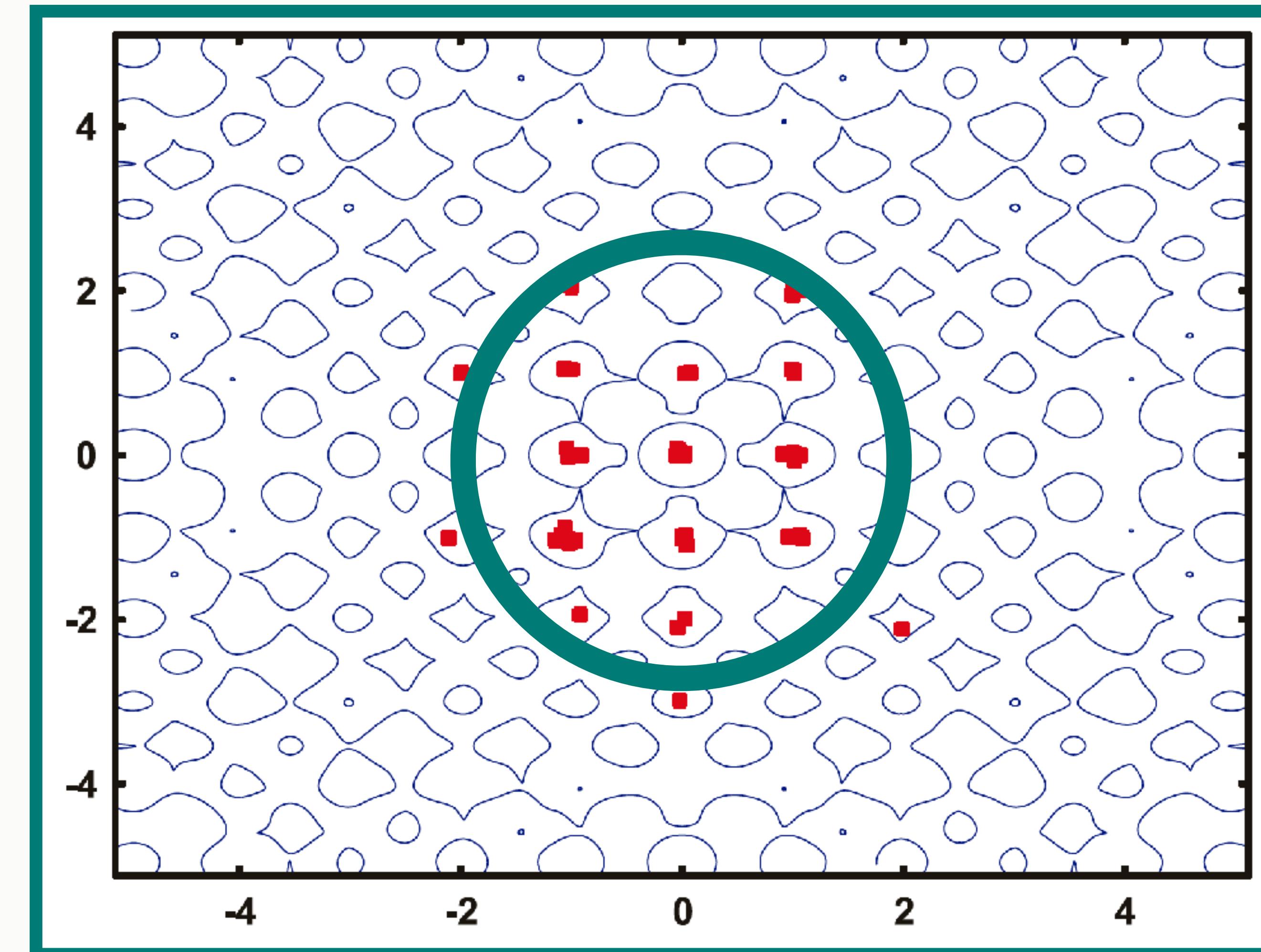
Generación
125



Evolución diferencial

ejemplo: minimización de función coseno

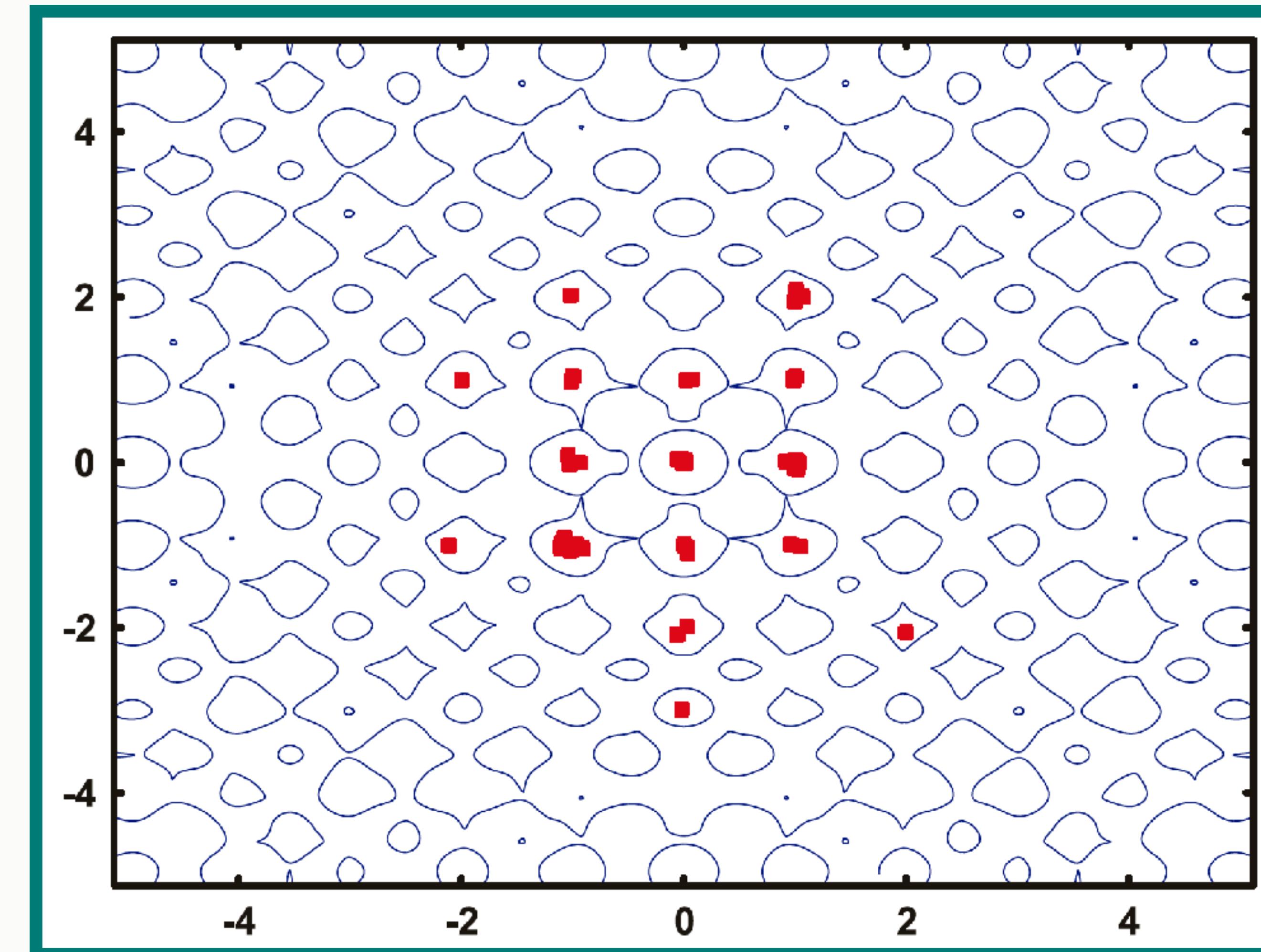
Generación
150



Evolución diferencial

ejemplo: minimización de función coseno

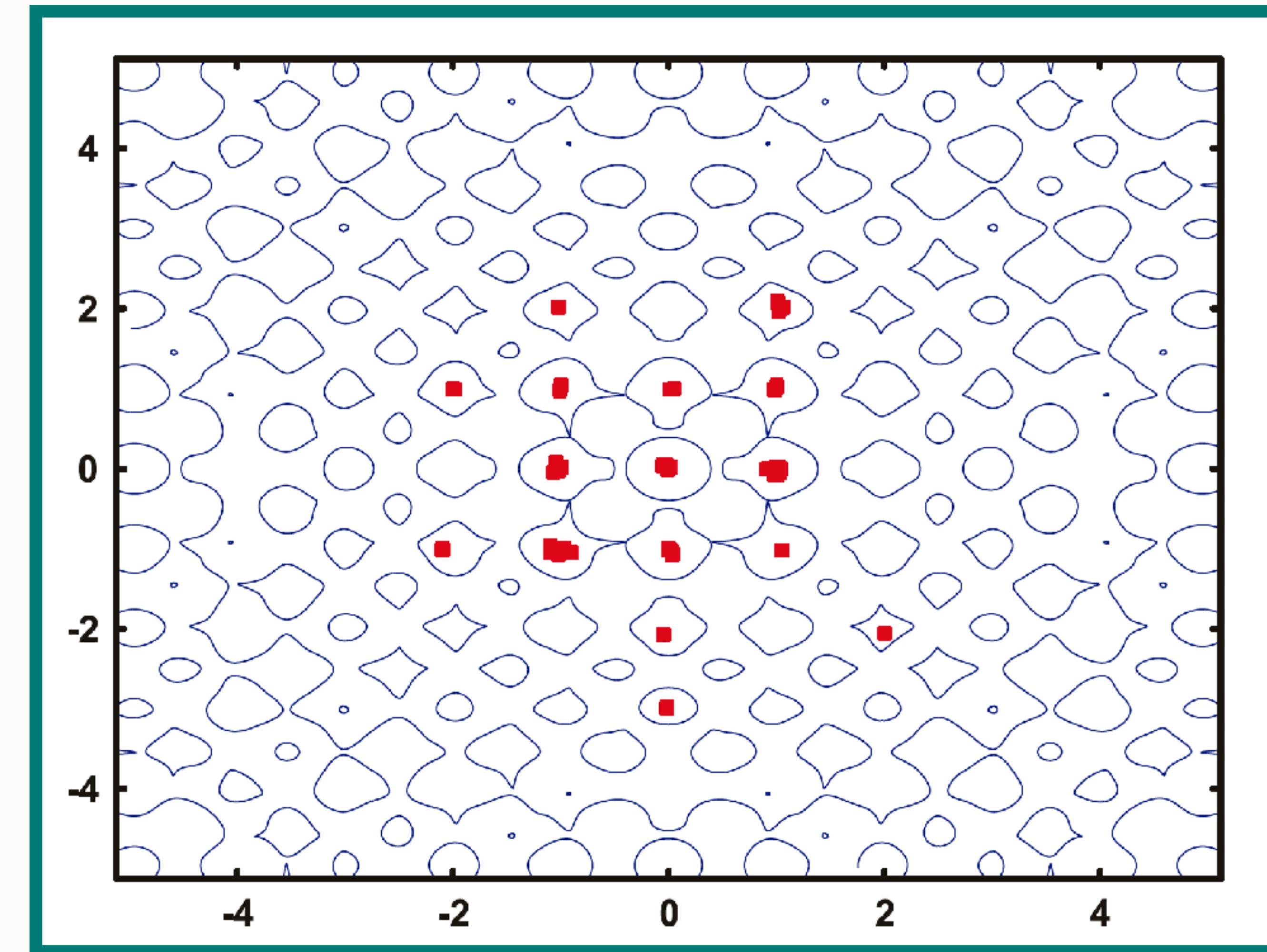
Generación
175



Evolución diferencial

ejemplo: minimización de función coseno

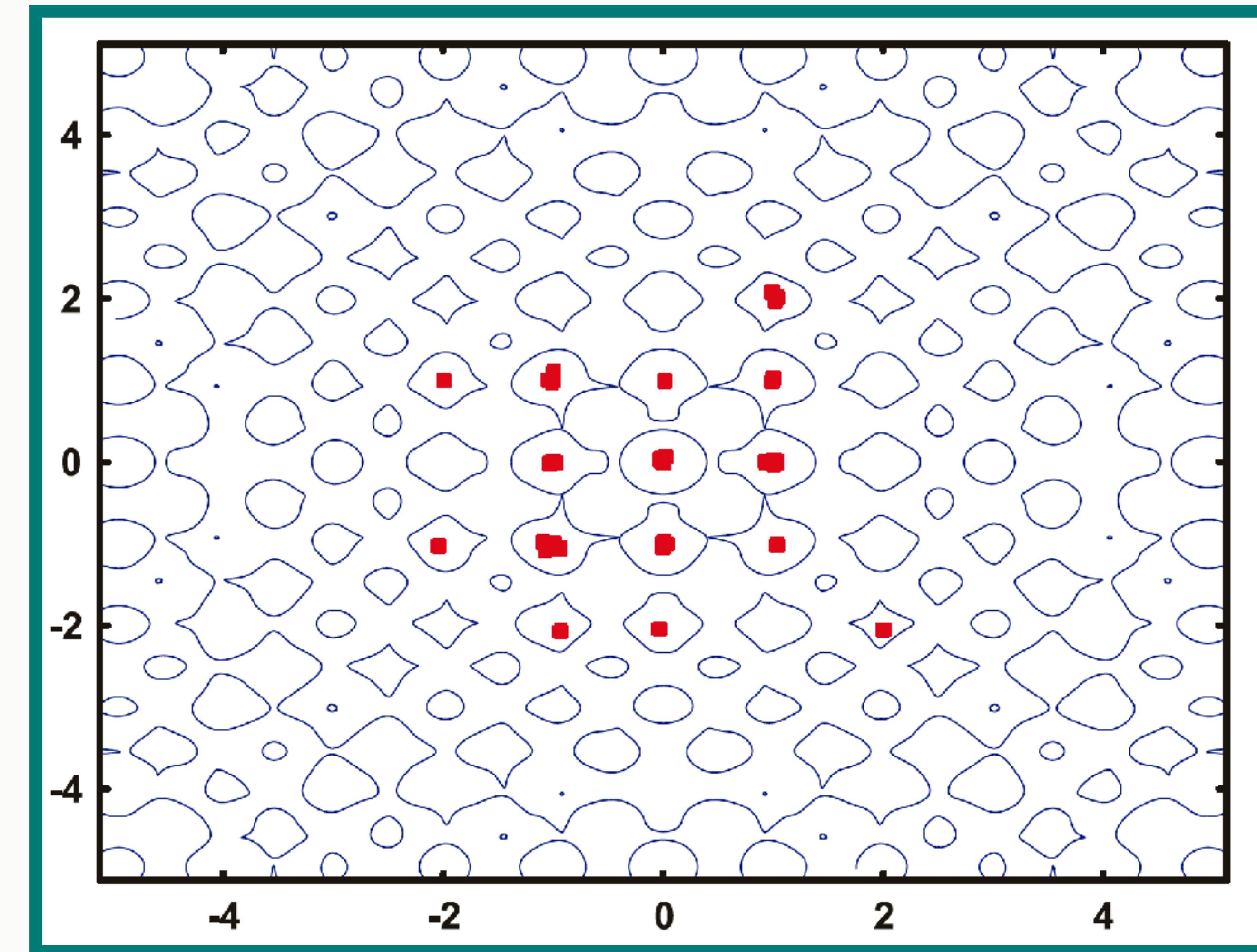
Generación
200



Evolución diferencial

ejemplo: minimización de función coseno

**Generación
250**

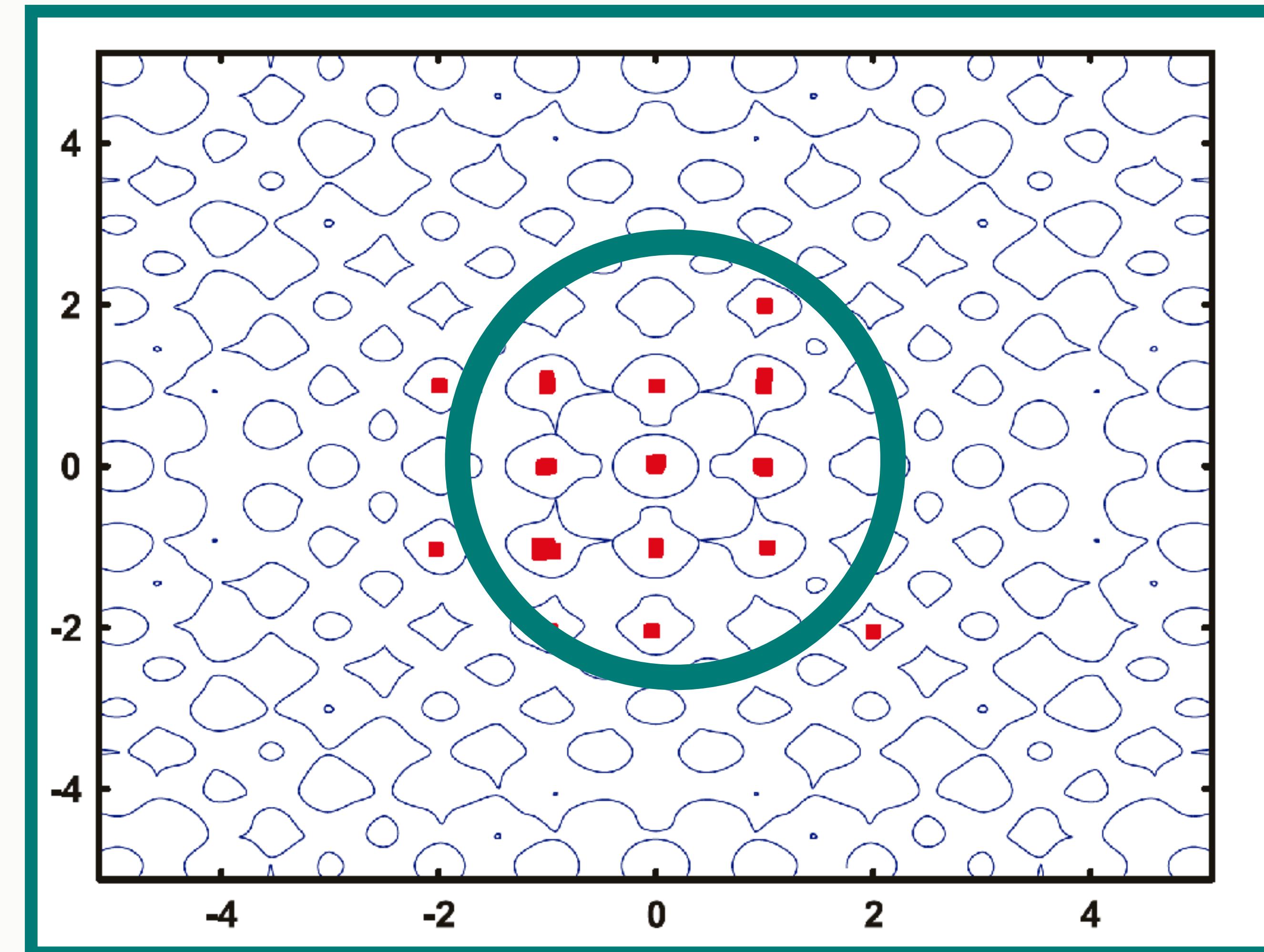


Evolución diferencial

ejemplo: minimización de función coseno

**Generación
300**

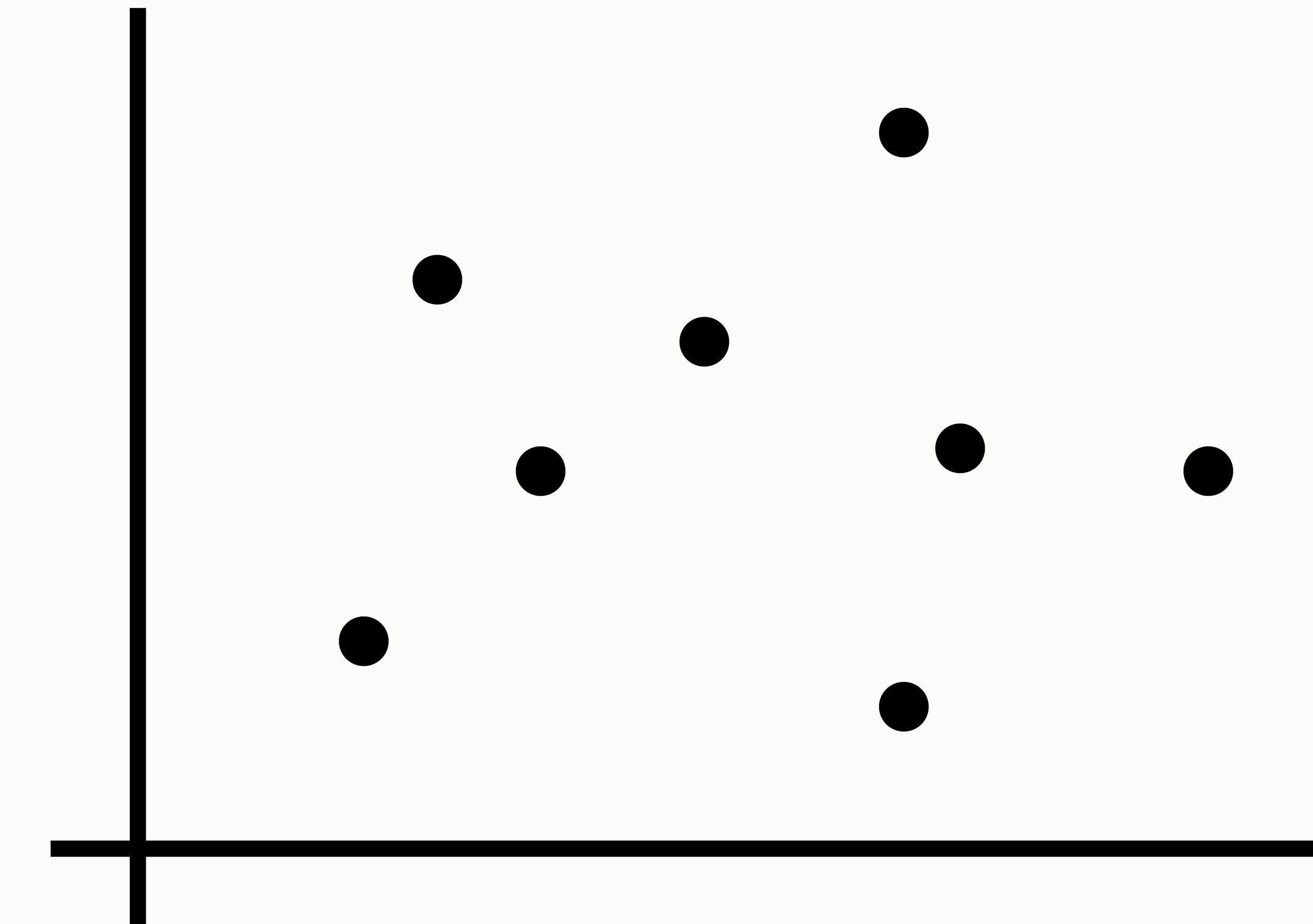
**Condición de
parada**



Evolución diferencial

operador de recombinación ternario

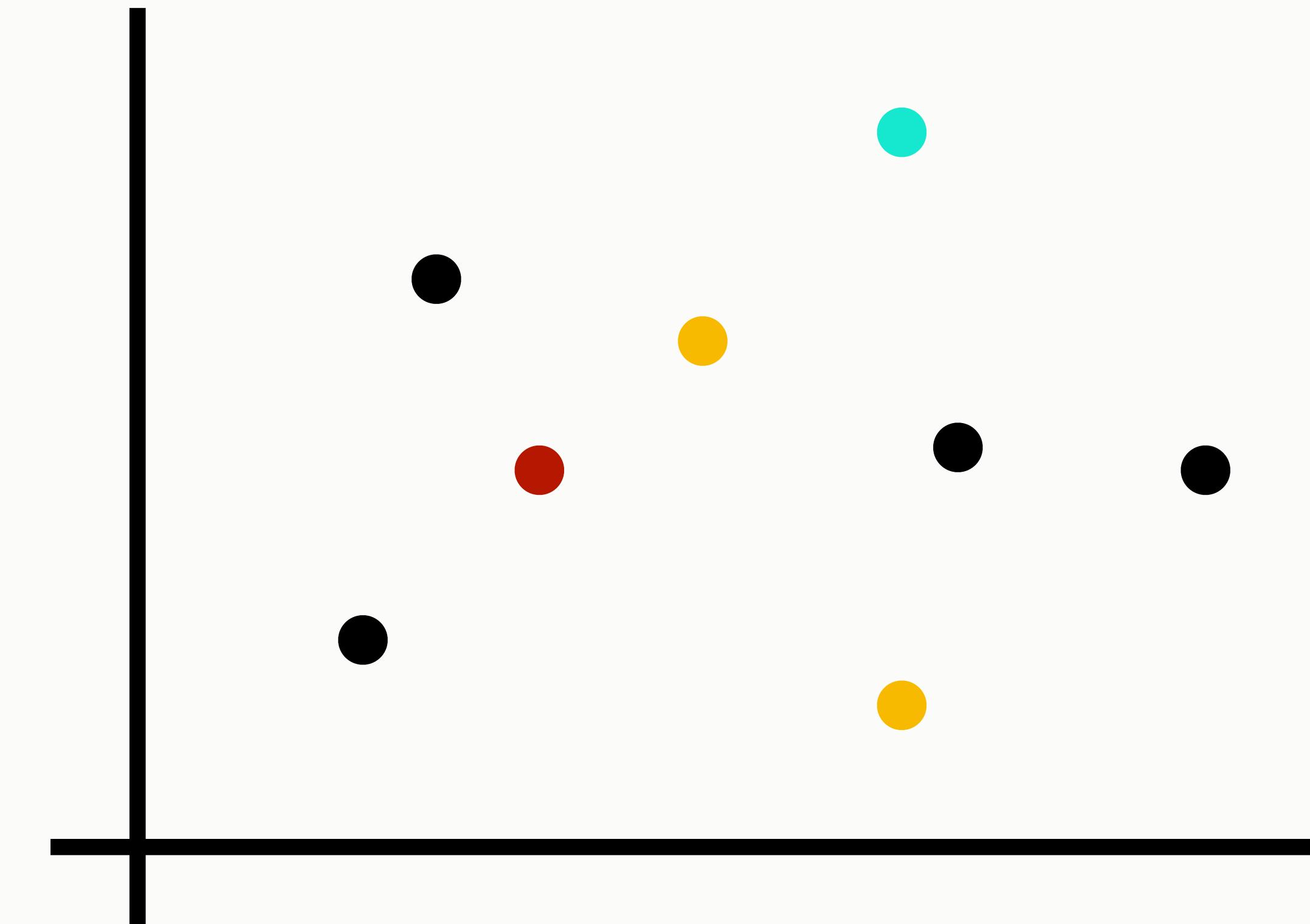
- Vamos a visualizar gráficamente cómo funciona este operador con un ejemplo clásico



Evolución diferencial

operador de recombinación ternario

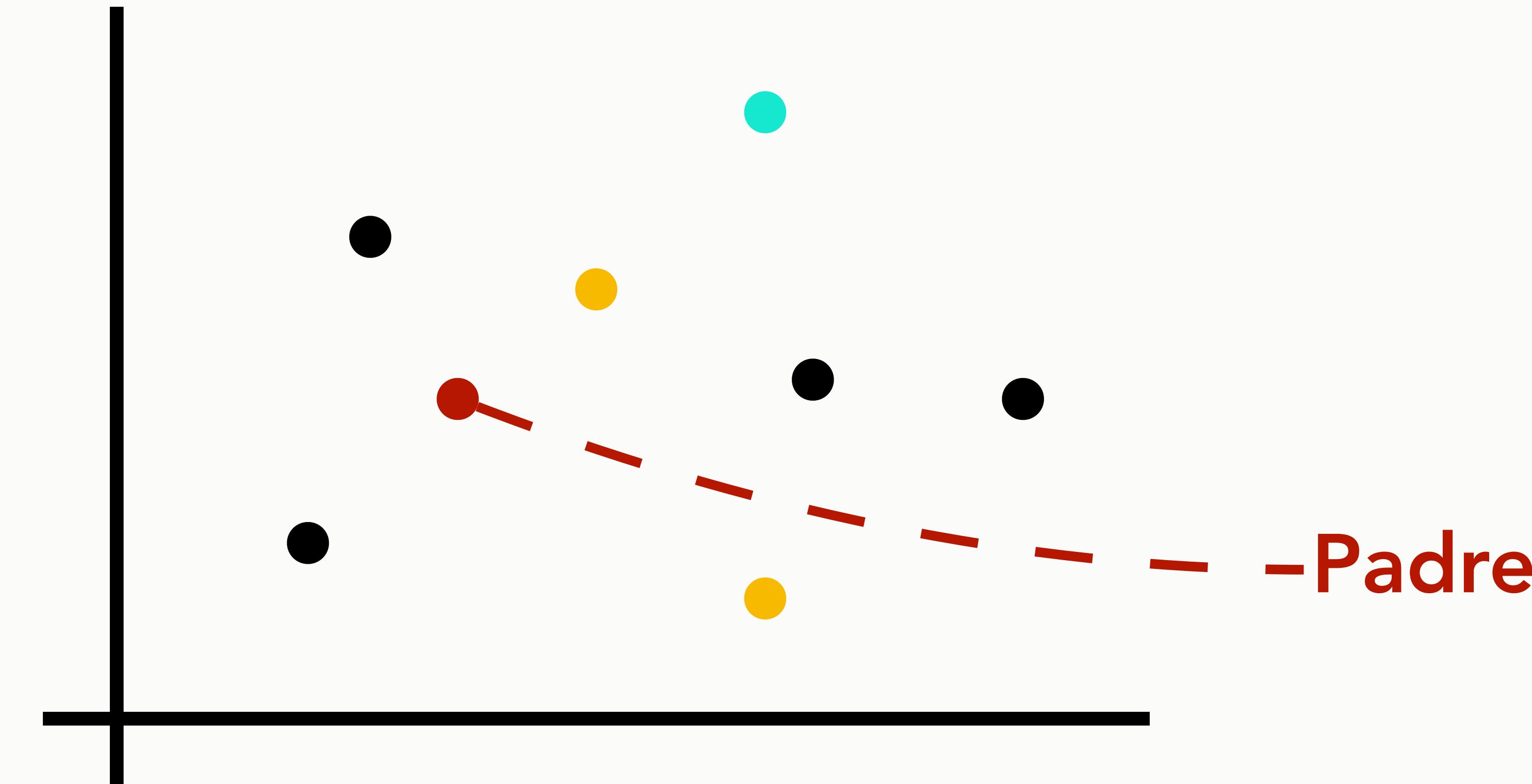
- Vamos a visualizar gráficamente cómo funciona este operador con un ejemplo clásico



Evolución diferencial

operador de recombinación ternario

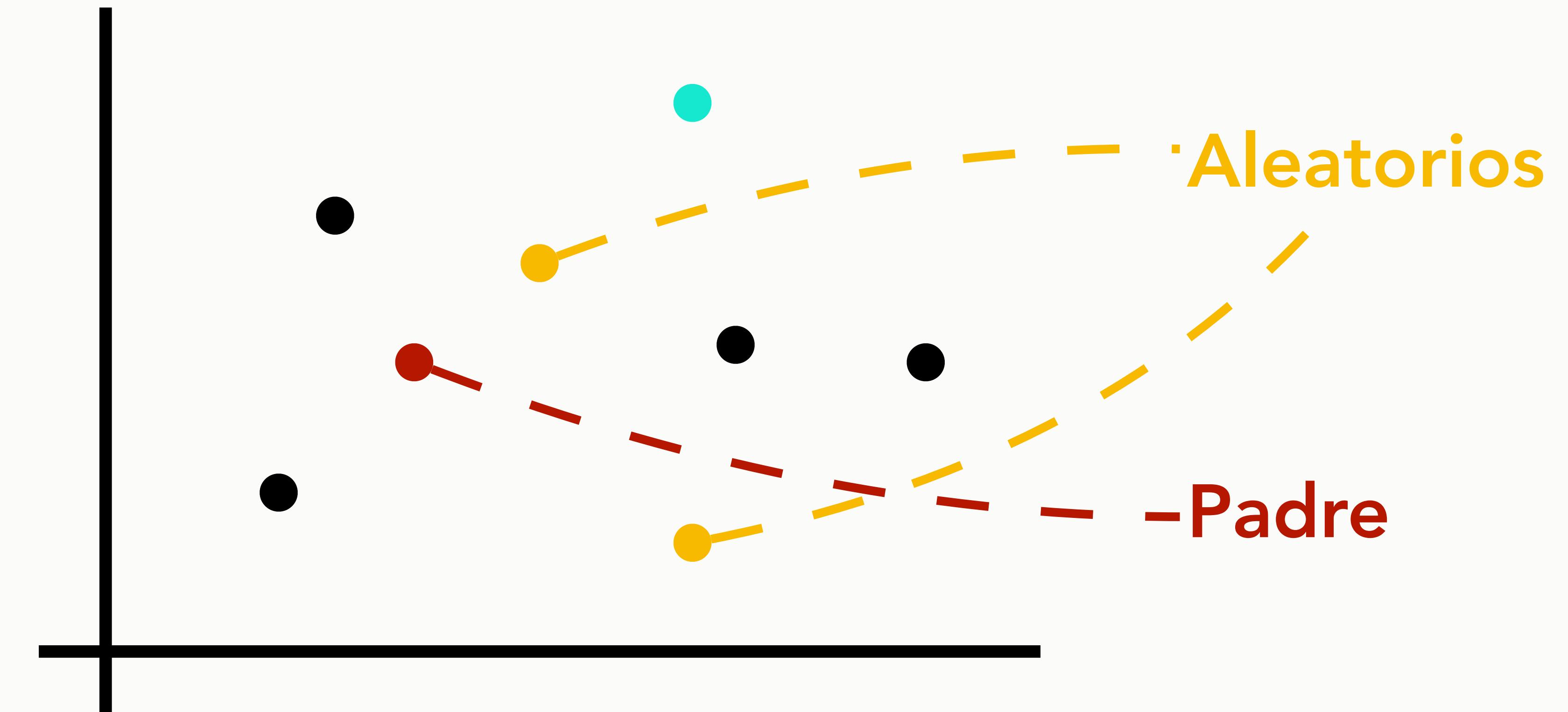
- Vamos a visualizar gráficamente cómo funciona este operador con un ejemplo clásico



Evolución diferencial

operador de recombinación ternario

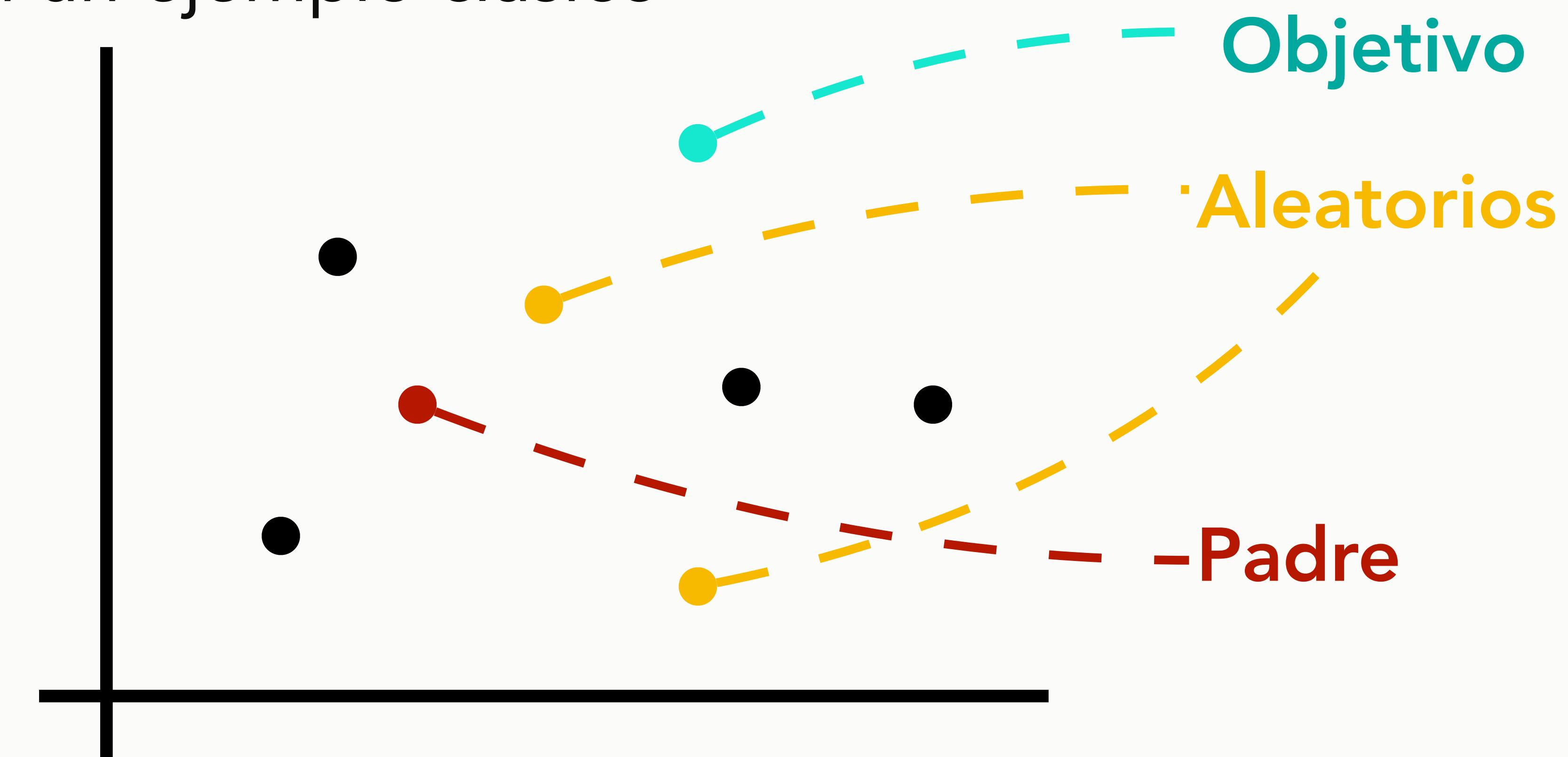
- Vamos a visualizar gráficamente cómo funciona este operador con un ejemplo clásico



Evolución diferencial

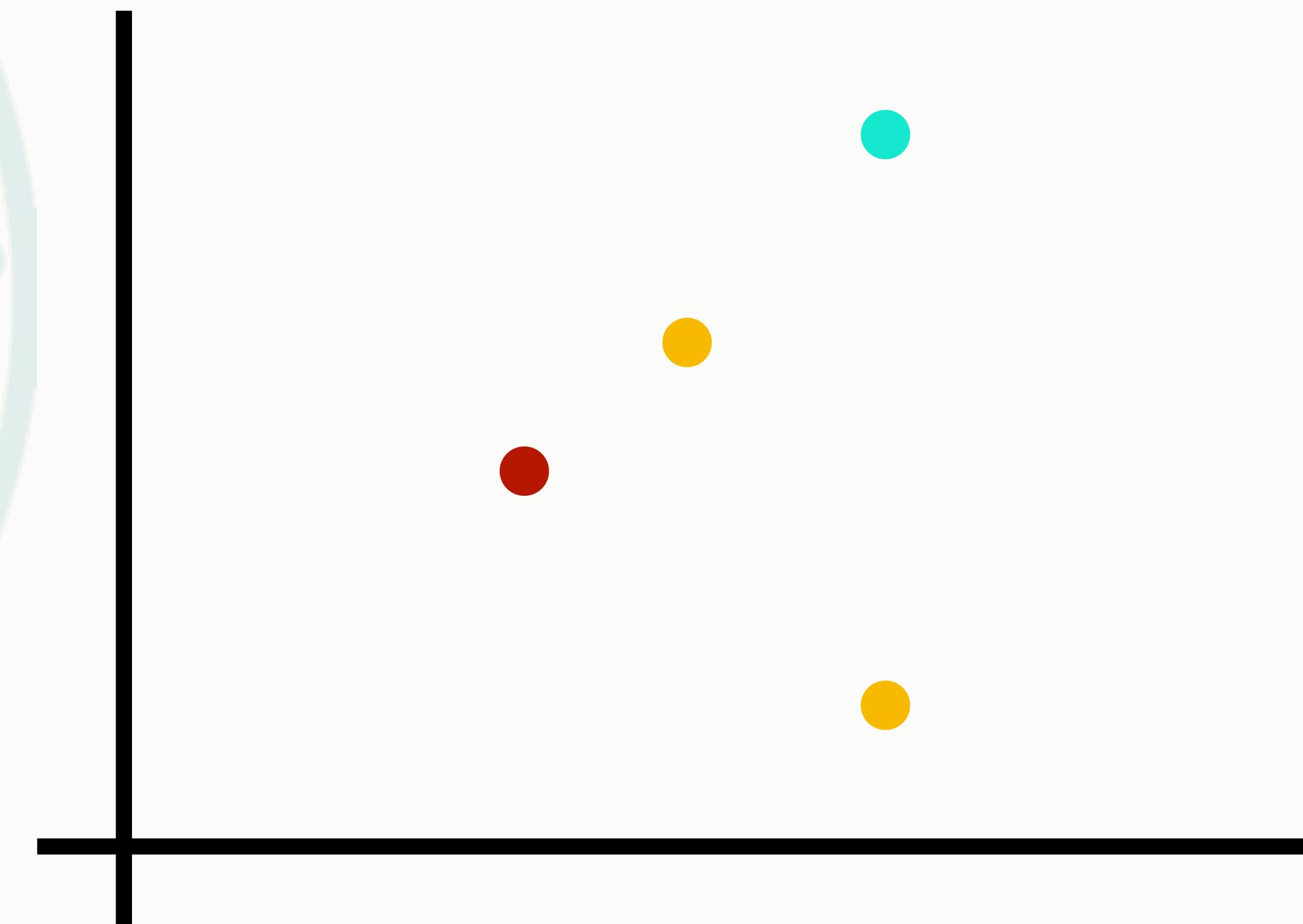
operador de recombinación ternario

- Vamos a visualizar gráficamente cómo funciona este operador con un ejemplo clásico



Evolución diferencial

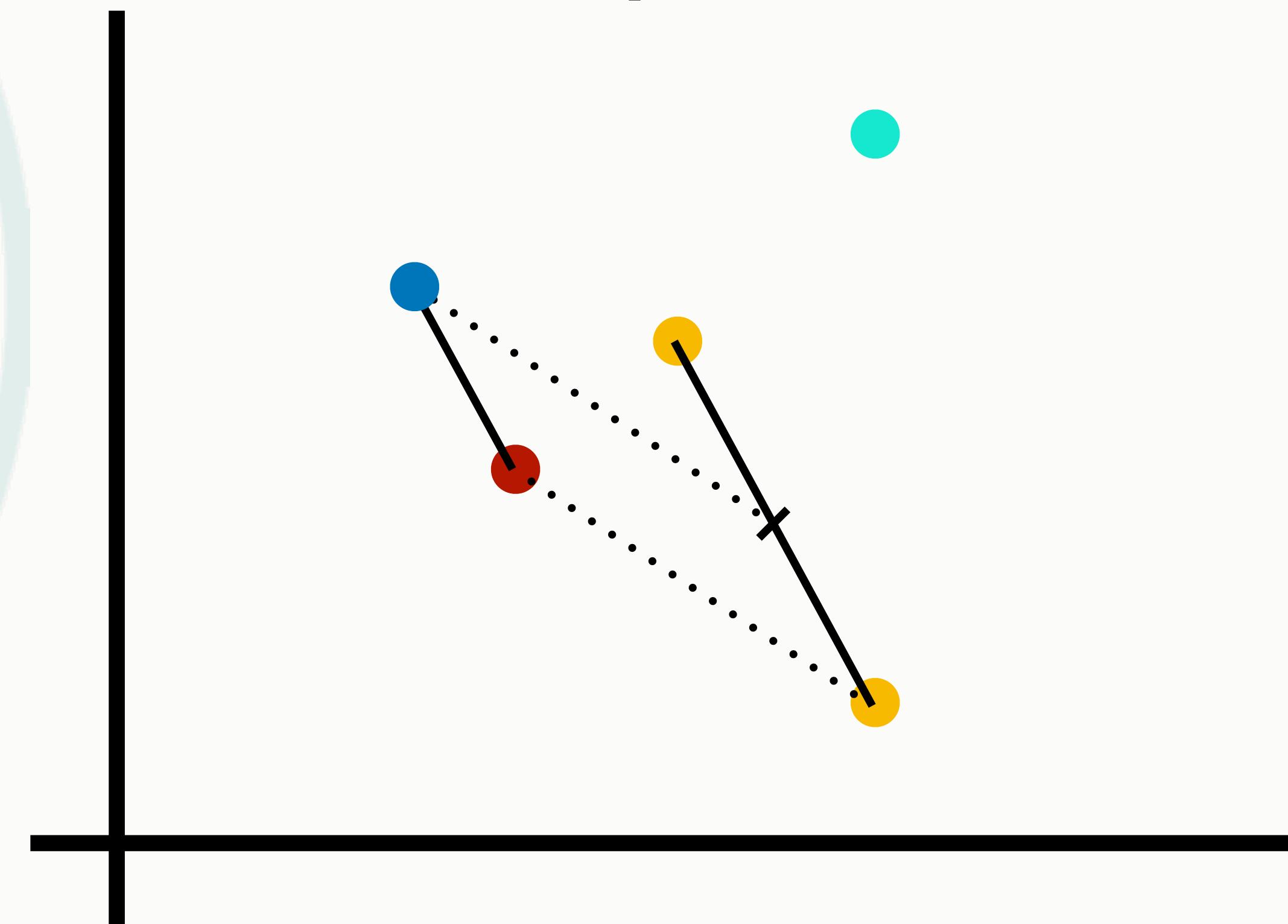
operador de recombinación ternario



Evolución diferencial

operador de recombinación ternario

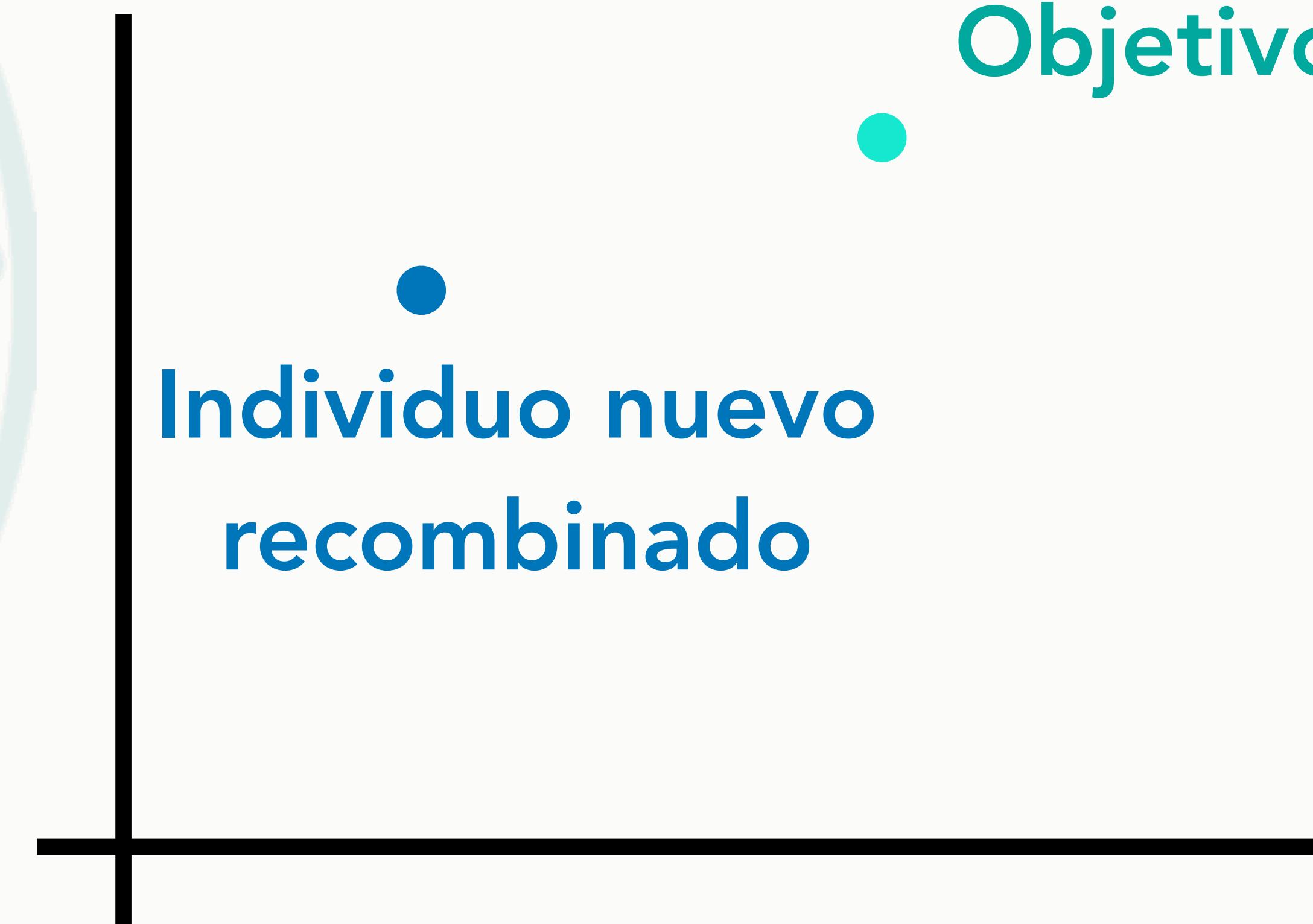
***F es la función de mutación
asociada al operador***



Objetivo
Aleatorios
Padre

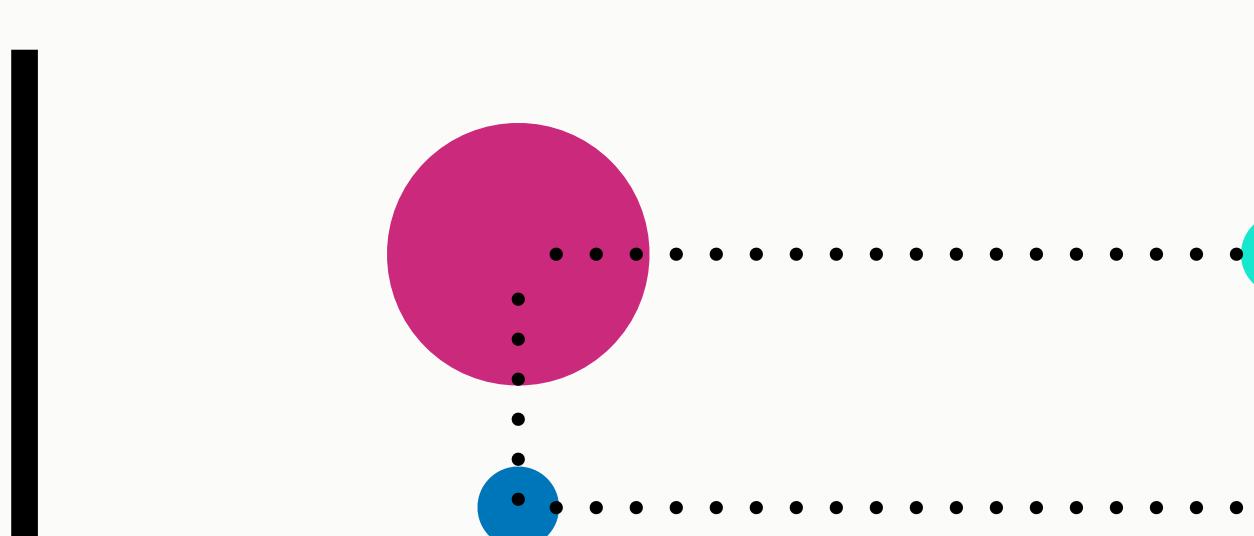
Evolución diferencial

operador de recombinación ternario



Evolución diferencial

operador de recombinación ternario



Individuo nuevo
recombinado

Objetivo

Por último, se realiza una recombinación discreta. Igual probabilidad para que aparezca el genotipo de ambos individuos

Evolución diferencial

operador de recombinación ternario

Solo hay reemplazo si mejora al Padre/Objetivo



Objetivo

Individuo nuevo
recombinado

Por último, se realiza una recombinación discreta. Igual probabilidad para que aparezca el genotipo de ambos individuos

Evolución diferencial

```
t = 0;  
Initialize Pop(t); Evaluate Pop(t);  
While (Not Done)  
    for i = 1 to |Pop(t)| do  
        parent1, parent2, parent3 = Select_3_Parents(Pop(t));  
        thisGene = random_int(|Pop(t)|);  
        for k = 1 to n do /* n genes per Individual */  
            if (random < p) Offspringik = parent1ik + γ(parent2ik – parent3ik);  
            else Offspringik = Individualik in Pop(t);  
        end  
        Evaluate(Offspringi);  
    end  
    Pop(t+1) = {j | Offspringj better_than Individualj} ∪ {k | Individualk better_than Offspringk};  
    t = t + 1;  
End While
```

Variantes de la evolución diferencial

Las variaciones más destacadas se centran en el operador que acabamos de ver:

- Vector 1-diferencia

$$F(x_1 - x_2)$$

- Vector 2-diferencias

$$F(x_1 - x_2) + F(x_3 - x_4)$$

- Factor de la mutación (F)

Variantes de la evolución diferencial

Las variaciones más destacadas se centran en el operador que acabamos de ver:

- Factor de la mutación (F)
 - Crucial: buen balance entre exploración y explotación
 - Dependencia de la dimensión: basada en rotaciones de los vectores
 - Aleatoria: basada en las diferencias de las distribuciones

Variantes de la evolución diferencial

También se pueden encontrar variantes en la función de recombinación:

- Recombinación discreta basada en uno o varios puntos, exponencial uniformes
- Recombinación aritmética basada en rotaciones de los espacios entre soluciones

Algoritmos avanzados

En la literatura podemos encontrar un amplio número de algoritmos avanzados muy interesantes:

- SaDE

A.K. Qin, V.L. Huang, P.N. Suganthan. Differential Evolution Algorithm with strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 13:2 (2009) 398-417

- DEGL

S. Das, A. Abraham, U.K. Chakraborty, A. Konar. Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Transactions on Evolutionary Computation*, 13:3 (2009) 526-553

- JADE

J. Zhang, A.C. Sanderson. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation*, 13:5 (2009) 945-958

Algoritmos avanzados

JADE

- Trabaja con un concepto de mutación generalizado del algoritmo

DE/current_to_best

$$v_{i,g} = x_{i,g} + F_i(x_{best,g} - x_{i,g}) + F_i(x_{r1,g} - x_{r2,g})$$

Algoritmos avanzados

JADE

- Trabaja con un concepto de mutación generalizado del algoritmo

DE/current_to_best

$$v_{i,g} = x_{i,g} + F_i(x_{best,g} - x_{i,g}) + F_i(x_{r1,g} - x_{r2,g})$$

DE/current_to_pbest

Algoritmos avanzados

JADE

- Trabaja con un concepto de mutación generalizado del algoritmo

DE/current_to_best

$$v_{i,g} = x_{i,g} + F_i(x_{best,g} - x_{i,g}) + F_i(x_{r1,g} - x_{r2,g})$$

DE/current_to_pbest

Emplea el concepto de la población con los p mejores, y el bestP se selecciona de forma aleatoria

$$v_{i,g} = x_{i,g} + F_i(x_{bestP,g} - x_{i,g}) + F_i(x_{r1,g} - x_{r2,g})$$

Algoritmos avanzados

JADE

- Conceptos de adaptación de los parámetros
 - ★ Función F tiene un valor en $[0, 1]$ que se adapta a lo largo de la ejecución
 - ★ Función CR para la recombinación le ocurre lo mismo

Algoritmos avanzados

JADE

TABLE I
PSEUDO CODE OF JADE WITH ARCHIVE

line# Procedure of JADE with Archive

```

01 Begin
02   Set  $\mu_{CR} = 0.5$ ;  $\mu_F = 0.5$ ;  $A = \emptyset$ 
03   Create a random initial population  $\{\mathbf{x}_{i,0} | i = 1, 2, \dots, NP\}$ 
04   For  $g = 1$  to  $G$ 
05      $S_F = \emptyset$ ;  $S_{CR} = \emptyset$ ;
06     For  $i = 1$  to  $NP$ 
07       Generate  $CR_i = \text{randn}_i(\mu_{CR}, 0.1)$ ,  $F_i = \text{randc}_i(\mu_F, 0.1)$ 
08       Randomly choose  $\mathbf{x}_{\text{best},g}^p$  as one of the  $100p\%$  best vectors
09       Randomly choose  $\mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from current population  $\mathbf{P}$ 
10      Randomly choose  $\tilde{\mathbf{x}}_{r2,g} \neq \mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from  $\mathbf{P} \cup A$ 
11       $\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$ 
12      Generate  $j_{\text{rand}} = \text{randint}(1, D)$ 
13      For  $j = 1$  to  $D$ 
14        If  $i = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
```

```

14          If  $j = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
15             $u_{j,i,g} = v_{j,i,g}$ 
16          Else
17             $u_{j,i,g} = x_{j,i,g}$ 
18          End If
19        End For
20        If  $f(\mathbf{x}_{i,g}) \leq f(\mathbf{u}_{i,g})$ 
21           $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$ 
22        Else
23           $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$ ;  $\mathbf{x}_{i,g} \rightarrow A$ ;  $CR_i \rightarrow S_{CR}$ ,  $F_i \rightarrow S_F$ 
24        End If
25      End for
26      Randomly remove solutions from  $A$  so that  $|A| \leq NP$ 
27       $\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$ 
28       $\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$ 
29    End for
30  End
```

Algoritmos avanzados

JADE

TABLE I
PSEUDO CODE OF JADE WITH ARCHIVE

line# Procedure of JADE with Archive

```

01 Begin
02   Set  $\mu_{CR} = 0.5$ ;  $\mu_F = 0.5$ ;  $A = \emptyset$ 
03   Create a random initial population  $\{\mathbf{x}_{i,0} | i = 1, 2, \dots, NP\}$ 
04   For  $g = 1$  to  $G$ 
05      $S_F = \emptyset$ ;  $S_{CR} = \emptyset$ ;
06     For  $i = 1$  to  $NP$ 
07       Generate  $CR_i = \text{randn}_i(\mu_{CR}, 0.1)$ ,  $F_i = \text{randc}_i(\mu_F, 0.1)$ 
08       Randomly choose  $\mathbf{x}_{\text{best},g}^p$  as one of the  $100p\%$  best vectors
09       Randomly choose  $\mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from current population  $\mathbf{P}$ 
10      Randomly choose  $\tilde{\mathbf{x}}_{r2,g} \neq \mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from  $\mathbf{P} \cup A$ 
11       $\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$ 
12      Generate  $j_{\text{rand}} = \text{randint}(1, D)$ 
13      For  $j = 1$  to  $D$ 
14        If  $i = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
```

```

14          If  $j = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
15             $u_{j,i,g} = v_{j,i,g}$ 
16          Else
17             $u_{j,i,g} = x_{j,i,g}$ 
18          End If
19        End For
20        If  $f(\mathbf{x}_{i,g}) \leq f(\mathbf{u}_{i,g})$ 
21           $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$ 
22        Else
23           $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$ ;  $\mathbf{x}_{i,g} \rightarrow A$ ;  $CR_i \rightarrow S_{CR}$ ,  $F_i \rightarrow S_F$ 
24        End If
25      End for
26      Randomly remove solutions from  $A$  so that  $|A| \leq NP$ 
27       $\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$ 
28       $\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$ 
29    End for
30  End
```

Algoritmos avanzados

JADE

TABLE I
PSEUDO CODE OF JADE WITH ARCHIVE

line# Procedure of JADE with Archive

```

01 Begin
02   Set  $\mu_{CR} = 0.5$ ;  $\mu_F = 0.5$ ;  $A = \emptyset$ 
03   Create a random initial population  $\{\mathbf{x}_{i,0} | i = 1, 2, \dots, NP\}$ 
04   For  $g = 1$  to  $G$ 
05      $S_F = \emptyset$ ;  $S_{CR} = \emptyset$ ;
06     For  $i = 1$  to  $NP$ 
07       Generate  $CR_i = \text{randn}_i(\mu_{CR}, 0.1)$ ,  $F_i = \text{randc}_i(\mu_F, 0.1)$ 
08       Randomly choose  $\mathbf{x}_{\text{best},g}^p$  as one of the  $100p\%$  best vectors
09       Randomly choose  $\mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from current population  $\mathbf{P}$ 
10      Randomly choose  $\tilde{\mathbf{x}}_{r2,g} \neq \mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from  $\mathbf{P} \cup A$ 
11       $\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$ 
12      Generate  $j_{\text{rand}} = \text{randint}(1, D)$ 
13      For  $j = 1$  to  $D$ 
14        If  $i = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
```

```

14          If  $j = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
15             $u_{j,i,g} = v_{j,i,g}$ 
16          Else
17             $u_{j,i,g} = x_{j,i,g}$ 
18          End If
19        End For
20        If  $f(\mathbf{x}_{i,g}) \leq f(\mathbf{u}_{i,g})$ 
21           $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$ 
22        Else
23           $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$ ;  $\mathbf{x}_{i,g} \rightarrow A$ ;  $CR_i \rightarrow S_{CR}$ ,  $F_i \rightarrow S_F$ 
24        End If
25      End for
26      Randomly remove solutions from  $A$  so that  $|A| \leq NP$ 
27       $\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$ 
28       $\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$ 
29    End for
30  End
```

Algoritmos avanzados

JADE

TABLE I
PSEUDO CODE OF JADE WITH ARCHIVE

line# Procedure of JADE with Archive

```

01 Begin
02   Set  $\mu_{CR} = 0.5$ ;  $\mu_F = 0.5$ ;  $A = \emptyset$ 
03   Create a random initial population  $\{\mathbf{x}_{i,0} | i = 1, 2, \dots, NP\}$ 
04   For  $g = 1$  to  $G$ 
05      $S_F = \emptyset$ ;  $S_{CR} = \emptyset$ ;
06     For  $i = 1$  to  $NP$ 
07       Generate  $CR_i = \text{randn}_i(\mu_{CR}, 0.1)$ ,  $F_i = \text{randc}_i(\mu_F, 0.1)$ 
08       Randomly choose  $\mathbf{x}_{\text{best},g}^p$  as one of the  $100p\%$  best vectors
09       Randomly choose  $\mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from current population  $\mathbf{P}$ 
10      Randomly choose  $\tilde{\mathbf{x}}_{r2,g} \neq \mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from  $\mathbf{P} \cup A$ 
11       $\mathbf{v}_{i,g} = \mathbf{x}_{i,g} - F_i \cdot (\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) - F_i \cdot (\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$ 
12      Generate  $j_{\text{rand}} = \text{randint}(1, D)$ 
13      For  $j = 1$  to  $D$ 
14        If  $i = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
```

```

14          If  $j = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
15             $u_{j,i,g} = v_{j,i,g}$ 
16          Else
17             $u_{j,i,g} = x_{j,i,g}$ 
18          End If
19        End For
20        If  $f(\mathbf{x}_{i,g}) \leq f(\mathbf{u}_{i,g})$ 
21           $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$ 
22        Else
23           $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$ ;  $\mathbf{x}_{i,g} \rightarrow A$ ;  $CR_i \rightarrow S_{CR}$ ,  $F_i \rightarrow S_F$ 
24        End If
25      End for
26      Randomly remove solutions from  $A$  so that  $|A| \leq NP$ 
27       $\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$ 
28       $\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$ 
29    End for
30  End
```

Algoritmos avanzados

JADE

TABLE I
PSEUDO CODE OF JADE WITH ARCHIVE

line# Procedure of JADE with Archive

```

01 Begin
02   Set  $\mu_{CR} = 0.5$ ;  $\mu_F = 0.5$ ;  $A = \emptyset$ 
03   Create a random initial population  $\{\mathbf{x}_{i,0} | i = 1, 2, \dots, NP\}$ 
04   For  $g = 1$  to  $G$ 
05      $S_F = \emptyset$ ;  $S_{CR} = \emptyset$ ;
06     For  $i = 1$  to  $NP$ 
07       Generate  $CR_i = \text{randn}_i(\mu_{CR}, 0.1)$ ,  $F_i = \text{randc}_i(\mu_F, 0.1)$ 
08       Randomly choose  $\mathbf{x}_{\text{best},g}^p$  as one of the  $100p\%$  best vectors
09       Randomly choose  $\mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from current population  $\mathbf{P}$ 
10      Randomly choose  $\tilde{\mathbf{x}}_{r2,g} \neq \mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from  $\mathbf{P} \cup A$ 
11       $\mathbf{v}_{i,g} = \mathbf{x}_{i,g} - F_i \cdot (\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) - F_i \cdot (\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$ 
12      Generate  $j_{\text{rand}} = \text{randint}(1, D)$ 
13      For  $j = 1$  to  $D$ 
14        If  $i = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
```

```

14          If  $j = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
15             $u_{j,i,g} = v_{j,i,g}$ 
16          Else
17             $u_{j,i,g} = x_{j,i,g}$ 
18          End If
19        End For
20        If  $f(\mathbf{x}_{i,g}) \leq f(\mathbf{u}_{i,g})$ 
21           $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$ 
22        Else
23           $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$ ;  $\mathbf{x}_{i,g} \rightarrow A$ ;  $CR_i \rightarrow S_{CR}$ ,  $F_i \rightarrow S_F$ 
24        End If
25      End for
26      Randomly remove solutions from  $A$  so that  $|A| \leq NP$ 
27       $\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$ 
28       $\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$ 
29    End for
30  End
```

Algoritmos avanzados

JADE

TABLE I
PSEUDO CODE OF JADE WITH ARCHIVE

line# Procedure of JADE with Archive

```

01 Begin
02   Set  $\mu_{CR} = 0.5$ ;  $\mu_F = 0.5$ ;  $A = \emptyset$ 
03   Create a random initial population  $\{\mathbf{x}_{i,0} | i = 1, 2, \dots, NP\}$ 
04   For  $g = 1$  to  $G$ 
05      $S_F = \emptyset$ ;  $S_{CR} = \emptyset$ ;
06     For  $i = 1$  to  $NP$ 
07       Generate  $CR_i = \text{randn}_i(\mu_{CR}, 0.1)$ ,  $F_i = \text{randc}_i(\mu_F, 0.1)$ 
08       Randomly choose  $\mathbf{x}_{\text{best},g}^p$  as one of the  $100p\%$  best vectors
09       Randomly choose  $\mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from current population  $\mathbf{P}$ 
10      Randomly choose  $\tilde{\mathbf{x}}_{r2,g} \neq \mathbf{x}_{r1,g} \neq \mathbf{x}_{i,g}$  from  $\mathbf{P} \cup \mathbf{A}$ 
11       $\mathbf{v}_{i,g} = \mathbf{x}_{i,g} - F_i \cdot (\mathbf{x}_{\text{best},g}^p - \mathbf{x}_{i,g}) - F_i \cdot (\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$ 
12      Generate  $j_{\text{rand}} = \text{randint}(1, D)$ 
13      For  $j = 1$  to  $D$ 
14        If  $i = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
```

```

14          If  $j = j_{\text{rand}}$  or  $\text{rand}(0, 1) < CR_i$ 
15             $u_{j,i,g} = v_{j,i,g}$ 
16          Else
17             $u_{j,i,g} = x_{j,i,g}$ 
18          End If
19        End For
20        If  $f(\mathbf{x}_{i,g}) \leq f(\mathbf{u}_{i,g})$ 
21           $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$ 
22        Else
23           $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$ ;  $\mathbf{x}_{i,g} \rightarrow \mathbf{A}$ ;  $CR_i \rightarrow S_{CR}$ ,  $F_i \rightarrow S_F$ 
24        End If
25      End for
26      Randomly remove solutions from  $\mathbf{A}$  so that  $|\mathbf{A}| \leq NP$ 
27       $\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$ 
28       $\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$ 
29    End for
30  End
```

Algoritmos avanzados

JADE

- Estudios comparativos

Test functions	Initial range
$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$
$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$
$f_3(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	$[-100, 100]^D$
$f_4(x) = \max_i \{ x_i \}$	$[-100, 100]^D$
$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$
$f_6(x) = \sum_{i=1}^D \lfloor x_i + 0.5 \rfloor^2$	$[-100, 100]^D$
$f_7(x) = \sum_{i=1}^D i x_i^4 + \text{rand}[0, 1)$	$[-1.28, 1.28]^D$
$f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } + D \cdot 418.98288727243369$	$[-500, 500]^D$
$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$

Test functions	Initial range
$f_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right)$ $- \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^D$
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$
$f_{12}(x) = \frac{\pi}{D} \{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ <p>where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and</p> $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^D$
$f_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	$[-50, 50]^D$

Algoritmos avanzados

JADE

TABLE IV

EXPERIMENTAL RESULTS OF 30-DIMENSIONAL PROBLEMS f_1-f_{13} , AVERAGED OVER 50 INDEPENDENT RUNS

	Gen	JADE w/o archive Mean (Std Dev)	JADE with archive Mean (Std Dev)	jDE Mean (Std Dev)	SaDE Mean (Std Dev)	DE/rand/1/bin Mean (Std Dev)	PSO Mean (Std Dev)
f_1	1500	1.8E-60 (8.4E-60)	1.3E-54 (9.2E-54)	2.5E-28 (3.5E-28)	4.5E-20 (6.9E-20)	9.8E-14 (8.4E-14)	9.6E-42 (2.7E-41)
f_2	2000	1.8E-25 (8.8E-25)	3.9E-22 (2.7E-21)	1.5E-23 (1.0E-23)	1.9E-14 (1.05E-14)	1.6E-09 (1.1E-09)	9.3E-21 (6.3E-20)
f_3	5000	5.7E-61 (2.7E-60)	6.0E-87 (1.9E-86)	5.2E-14 (1.1E-13)	9.0E-37 (5.43E-36)	6.6E-11 (8.8E-11)	2.5E-19 (3.9E-19)
f_4	5000	8.2E-24 (4.0E-23)	4.3E-66 (1.2E-65)	1.4E-15 (1.0E-15)	7.4E-11 (1.82E-10)	4.2E-01 (1.1E+00)	4.4E-14 (9.3E-14)
f_5	3000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	1.3E+01 (1.4E+01)	2.1E+01 (7.8E+00)	2.1E+00 (1.5E+00)	2.5E+01 (3.2E+01)
	20000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	8.0E-02 (5.6E-01)	1.8E+01 (6.7E+00)	8.0E-02 (5.6E-01)	1.7E+01 (2.3E+01)
f_6	100	2.9E+00 (1.2E+00)	5.6E+00 (1.6E+00)	1.0E+03 (2.2E+02)	9.3E+02 (1.8E+02)	4.7E+03 (1.1E+03)	4.5E+01 (2.4E+01)
	1500	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	8.0E-02 (2.7E-01)
f_7	3000	6.4E-04 (2.5E-04)	6.8E-04 (2.5E-04)	3.3E-03 (8.5E-04)	4.8E-03 (1.2E-03)	4.7E-03 (1.2E-03)	2.5E-03 (1.4E-03)
f_8	1000	3.3E-05 (2.3E-05)	7.1E+00 (2.8E+01)	7.9E-11 (1.3E-10)	4.7E+00 (3.3E+01)	5.9E+03 (1.1E+03)	2.4E+03 (6.7E+02)
	9000	0.0E+00 (0.0E+00)	7.1E+00 (2.8E+01)	0.0E+00 (0.0E+00)	4.7E+00 (3.3E+01)	5.7E+01 (7.6E+01)	2.4E+03 (6.7E+02)
f_9	1000	1.0E-04 (6.0E-05)	1.4E-04 (6.5E-05)	1.5E-04 (2.0E-04)	1.2E-03 (6.5E-04)	1.8E+02 (1.3E+01)	5.2E+01 (1.6E+01)
	5000	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	7.1E+01 (2.1E+01)	5.2E+01 (1.6E+01)
f_{10}	500	8.2E-10 (6.9E-10)	3.0E-09 (2.2E-09)	3.5E-04 (1.0E-04)	2.7E-03 (5.1E-04)	1.1E-01 (3.9E-02)	4.6E-01 (6.6E-01)
	2000	4.4E-15 (0.0E+00)	4.4E-15 (0.0E+00)	4.7E-15 (9.6E-16)	4.3E-14 (2.6E-14)	9.7E-11 (5.0E-11)	4.6E-01 (6.6E-01)
f_{11}	500	9.9E-08 (6.0E-07)	2.0E-04 (1.4E-03)	1.9E-05 (5.8E-05)	7.8E-04 (1.2E-03)	2.0E-01 (1.1E-01)	1.3E-02 (1.7E-02)
	3000	0.0E+00 (0.0E+00)	2.0E-04 (1.4E-03)	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	0.0E+00 (0.0E+00)	1.1E-02 (1.6E-02)
f_{12}	500	4.6E-17 (1.9E-16)	3.8E-16 (8.3E-16)	1.6E-07 (1.5E-07)	1.9E-05 (9.2E-06)	1.2E-02 (1.0E-02)	1.9E-01 (3.9E-01)
	1500	1.6E-32 (5.5E-48)	1.6E-32 (5.5E-48)	2.6E-29 (7.5E-29)	1.2E-19 (2.0E-19)	1.1E-14 (1.0E-14)	1.9E-01 (3.9E-01)
f_{13}	500	2.0E-16 (6.5E-16)	1.2E-15 (2.8E-15)	1.5E-06 (9.8E-07)	6.1E-05 (2.0E-05)	7.5E-02 (3.8E-02)	2.9E-03 (4.8E-03)
	1500	1.4E-32 (1.1E-47)	1.4E-32 (1.1E-47)	1.9E-28 (2.2E-28)	1.7E-19 (2.4E-19)	7.5E-14 (4.8E-14)	2.9E-03 (4.8E-03)

Comentarios finales

Los algoritmos de Evolución Diferencial son uno de los campos más activos en el desarrollo de algoritmos evolutivos para la optimización de parámetros (optimización continua)

La hibridación de los algoritmos de Evolución Diferencial y los algoritmos de búsqueda local son una vía de trabajo con potenciales buenos resultados en el ámbito de la optimización continua



Metaheurísticas
Grado en Ingeniería Informática
Universidad de Jaén
Cristóbal J. Carmona
Curso 2023/2024

Esta obra está protegida con licencia
Creative Commons Atribución-NoComercial 4.0 Internacional

