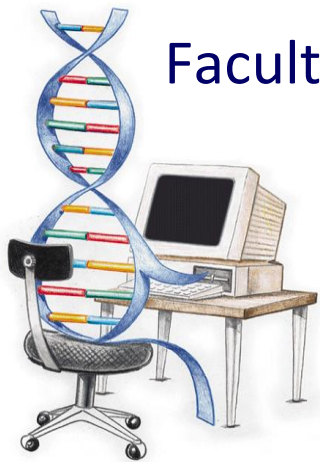


ALGORITMOS EVOLUTIVOS

Curso 2024

Tema 8: Técnicas avanzadas en AE

Centro de Cálculo, Instituto de Computación
Facultad de Ingeniería, Universidad de la República, Uruguay



cecal



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



Contenido

- Codificación diploide
- Inversión y reordenamiento
- Representación de permutaciones
- Mecanismos para preservar la diversidad
 - Diferenciación sexual, Especialización y nichos, Técnicas de crowding, Fitness sharing, Restricciones al cruzamiento



Codificación diploide

- El AG simple utiliza una representación haploide: un único alelo por cada gen del cromosoma. Por ejemplo:

01110

- En una representación diploide cada genotipo tiene dos cromosomas asociados. Por ejemplo:

01110

10111

- Ambos cromosomas proporcionan información sobre la solución representada.
- En la naturaleza muchos organismos tienen cromosomas con estructura diploide.

Codificación diploide

- Existe redundancia en la codificación cromosómica. Es necesario un mecanismo de desambiguación.
- Dominancia
 - Permite determinar qué alelos serán expresados.
 - Es decir, qué características de las que se representan estarán presentes en el fenotipo.
 - En cada locus, algunos alelos dominan (dominantes) sobre otros (recesivos).
 - Los valores dominantes se manifiestan en el fenotipo, mientras que los recesivos no se manifiestan.

Codificación diploide

- Por ejemplo, si se considera como criterio para cada uno de los locus la dominancia de mayúsculas

AbCdE

abCDE

El fenotipo tendrá las características AbCDE.

- Las características dominantes se expresan en homocigotos ($AA \rightarrow A$) y heterocigotos ($Aa \rightarrow A$).
- Las características recesivas solamente se expresan en homocigotos ($aa \rightarrow a$).

Codificación diploide

¿Por qué introducir redundancia mediante diploides?

- Proveen un mecanismo de memoria para recordar valores que fueron útiles en el pasado.
- Son útiles para problemas con funciones de fitness variables que modelan entornos adaptativos.
 - Mantiene información que puede ser utilizada en el futuro.
- Permite mantener múltiples soluciones diferentes con una misma solución expresada, aumentando así la diversidad de la población.

Codificación diploide: aplicación en AE

- La dominancia puede ser estática o dinámica. Para el caso dinámico existen diferentes enfoques.
- *Dominancia global variable*: la probabilidad de dominancia de un alelo se calcula según su proporción en la generación actual.
 - *Dominancia global variable determinista*: siempre domina el alelo con mayor proporción.
- *Basado en fitness*: el alelo más efectivo en términos de fitness promedio se transforme en dominante y el otro en recesivo.
- *Dominancia basada en haploide*: se usa un cromosoma haploide extra asociado a cada cromosoma diploide para codificar la dominancia.

Inversión y reordenamiento

- La inversión es un mecanismo utilizado para reducir el sesgo de los operadores de cruzamiento de n puntos.
- Funciona simplemente invirtiendo el orden de los elementos del genotipo entre dos posiciones.
- Ejemplo:
 $0\ 1\ 0\ 0\ 1\ 1 \rightarrow 0\ 1\ 1\ 0\ 0\ 1$
- Su objetivo es reordenar las posiciones de la codificación acercando posiciones potencialmente relacionadas.

Inversión y reordenamiento

- Se propone tomar en cuenta la posición asociándola al alelo correspondiente mediante una codificación (posición, alelo).
- Ejemplo:
 $(1, \underline{0})(2, \underline{1})(3, \underline{0})(4, \underline{0})(5, \underline{1})(6, \underline{1}) \rightarrow (1, \underline{0})(2, \underline{1})(5, \underline{1})(4, \underline{0})(3, \underline{0})(6, \underline{1})$
- Se modifica el *genotipo* manteniendo el *fenotipo* incambiado.
- Desventaja: no es posible realizar el cruzamiento en forma directa.

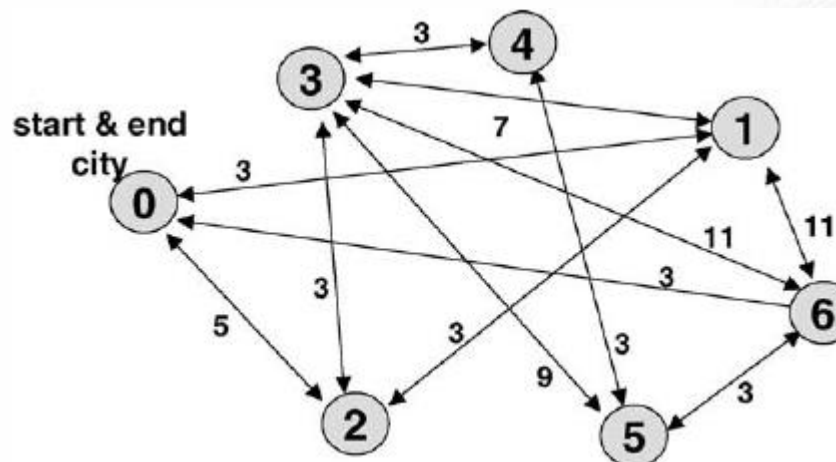
Representación de permutaciones

- Es utilizada en problemas de optimización combinatoria donde se busca una permutación que optimice una cierta función objetivo.
- Ejemplo: El problema del vendedor viajero (TSP). Cada entero se corresponde con un identificador de una ciudad en un ciclo solución.

- Ejemplo:

- 0 1 2 3 4 5 6

- 0 2 3 4 5 6 1



- Los operadores tradicionales no pueden ser aplicados trivialmente, es necesario utilizar operadores específicos.



Representación de permutaciones



- Operadores de mutación
 - Exchange Mutation
 - Insertion Mutation
 - Displacement Mutation
 - Simple Inversion Mutation
 - Inversion Mutation
- Operadores de cruzamiento
 - Partially Mapped Crossover
 - Order Crossover
 - Cycle Crossover
 - Position Based Crossover

Mutación para la representación de permutaciones

- EM – Exchange Mutation

Se sortean dos posiciones dentro del individuo y se intercambian los valores en dichas posiciones.

$$A = 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \quad \Rightarrow \quad A' = 1\ 2\ 7\ 4\ 5\ 6\ 3\ 8\ 9$$

- ISM – Insertion Mutation

Se sortea una posición y se “mueve” a una posición aleatoria.

$$A = 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \quad \Rightarrow \quad A' = 1\ 2\ 6\ 3\ 4\ 5\ 7\ 8\ 9$$

- DM – Displacement Mutation

Igual a ISM pero con un “grupo” de genes.

$$A = 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \quad \Rightarrow \quad A' = 1\ 5\ 6\ 7\ 2\ 3\ 4\ 8\ 9$$

Mutación para la representación de permutaciones

- *SIM – Simple Inversion Mutation*

Se sortea una sección del individuo y se invierte el orden de los valores en los genes de la sección

$$A = 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \quad \Rightarrow \quad A' = 1 \ 4 \ 3 \ 2 \ 5 \ 6 \ 7 \ 8 \ 9$$

- *IM – Inversion Mutation*

Combina inversión simple y desplazamiento (SIM + DM)

$$A = 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \quad \Rightarrow \quad A' = 1 \ 5 \ 6 \ 7 \ 4 \ 3 \ 2 \ 8 \ 9$$

Cruzamiento para la representación de permutaciones

- PMX - Partially Mapped Crossover

Actúa intercambiando secciones entre dos puntos de corte (como 2PX), pero corrige en base al mapeo entre padres.

Ejemplo, cruce entre individuos A y B:

A = 1 2 3 | 4 5 6 | 7 8 9 ⇒ 1 2 3 | 2 9 7 | 7 8 9
B = 8 6 4 | 2 9 7 | 5 3 1 ⇒ 8 6 4 | 4 5 6 | 5 3 1

La sección intercambiada permanece igual. El *resto de la permutación* se corrige usando el mapeo de la parte intercambiada:

2	9	7
↕	↕	↕
4	5	6

Cruzamiento para la representación de permutaciones

- PMX - Partially Mapped Crossover (continuación)

- Individuos corregidos:

1 2 3 | 2 9 7 | 7 8 9 ⇒ 1 4 3 | 2 9 7 | 6 8 5
8 6 4 | 4 5 6 | 5 3 1 8 7 2 | 4 5 6 | 9 3 1

- La idea que guía a este operador es mantener las *posiciones absolutas* de los elementos codificados en los individuos.

Cruzamiento para la representación de permutaciones

- OX – Order Crossover

Es similar al PMX, pero no corrige por intercambios sino que genera “huecos” que se desplazan en orden circular a partir del segundo punto de corte. Los huecos se mapean con el otro individuo.

Ejemplo, cruce entre individuos A y B:

A = 1 2 3 | 4 5 6 | 7 8 9

B = 8 6 4 | 2 9 7 | 5 3 1

Para el individuo A resulta:

1 **H** 3 | 4 5 6 | **H** 8 **H**

4 5 6 | **H** **H** **H** | 8 1 3

4 5 6 | **2** **9** **7** | 8 1 3

Cruzamiento para la representación de permutaciones

- OX – Order Crossover

Para el individuo B resulta:

8	H	H		2	9	7		H	3	1
2	9	7		H	H	H		3	1	8
2	9	7		4	5	6		3	1	8

En este caso el operador de cruzamiento está orientado a mantener las *posiciones relativas de los elementos* codificados en el individuo, siguiendo un orden cíclico.

Cruzamiento para la representación de permutaciones

- CX – Cycle Crossover

Se selecciona un valor en un individuo y se completa con los mapeos correspondientes en el otro individuo, hasta obtener un ciclo. Posteriormente se completa con los valores “sobrantes” del otro individuo.

Ejemplo, cruce entre individuos A y B:

A = 9 8 2 1 7 4 5 3 6

B = 1 2 3 4 5 6 7 8 9

Para el primer individuo:

9 – – 1 – 4 – – 6 se completa el mapeo hasta obtener un ciclo.

9 2 3 1 5 4 7 8 6 se completa con los valores del otro individuo.

Este operador de cruzamiento *mantiene los ciclos*.

Cruzamiento para la representación de permutaciones

- PBX – Postion Based Crossover

1. Seleccionar varios valores del padre 1 que se copian directamente al hijo 1 en las mismas posiciones.
2. Marcar en el padre 2 los valores ya copiados.
3. Desde el comienzo del padre 2, copiar todos los valores no marcados en las posiciones libres del hijo 1 empezando por el comienzo.

Ejemplo, cruce entre individuos A y B:

A = 9 8 2 1 7 4 5 3 6

B = 1 2 3 4 5 6 7 8 9

Para el primer individuo:

9 - 2 - - 4 5 3 - se copian los valores elegidos del padre 1.

9 1 2 6 7 4 5 3 8 se completa con los valores del padre 2.

Mecanismos para mantener la diversidad

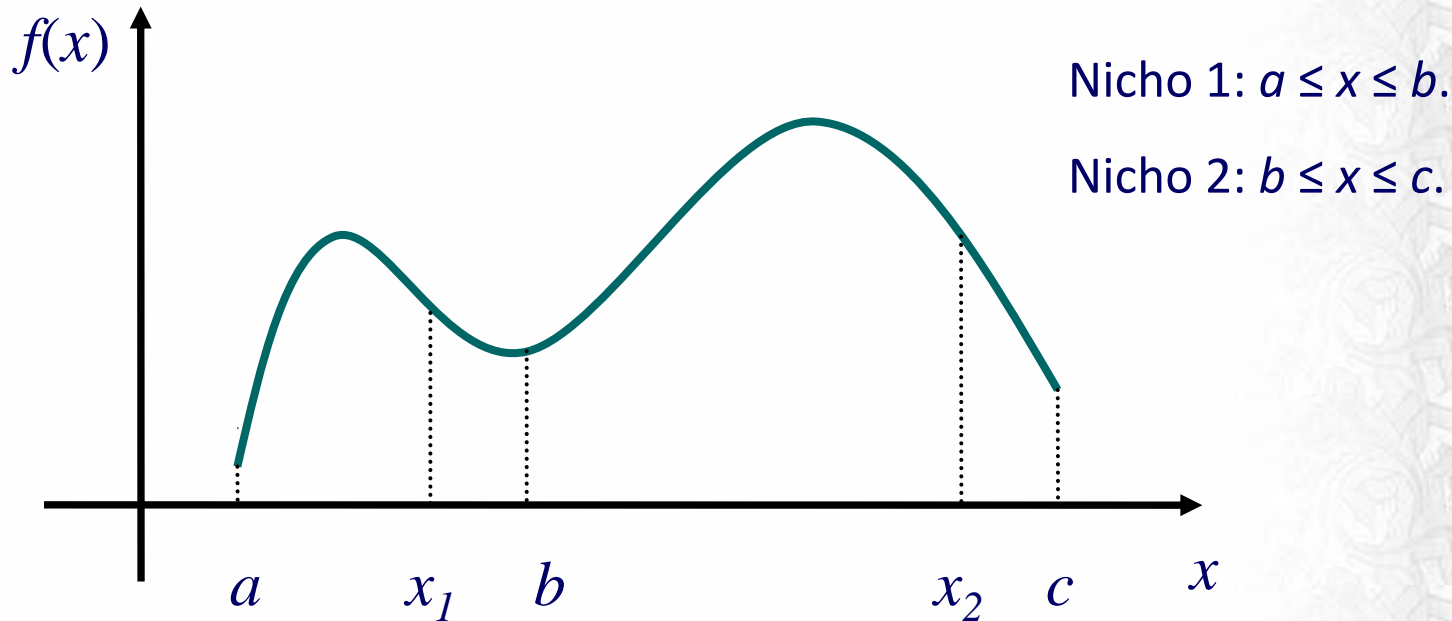
- La **diversidad genética** es un concepto muy importante para alcanzar resultados de calidad cuando se trabaja con algoritmos evolutivos.
- Algunos mecanismos de preservación de la diversidad:
 - Especiación y nichos
 - Técnicas de crowding
 - Fitness sharing
 - Restricciones al cruzamiento
- Su objetivo es **preservar la mayor cantidad de características de los individuos** presentes en la población.

Especiación y nichos

- *Especiación*: separación de individuos en clases.
- *Nicho*: trabajo o rol que realiza un determinado organismo.
- Son mecanismos basados en la definición de categorías (subpoblaciones, castas, etc.) y *restricciones en el cruzamiento*.
- Intentan preservar características de ciertos individuos para muestrear adecuadamente el espacio de búsqueda.
- Se utilizan en la optimización de funciones multimodales y para problemas con fitness variable.

Especiación y nichos

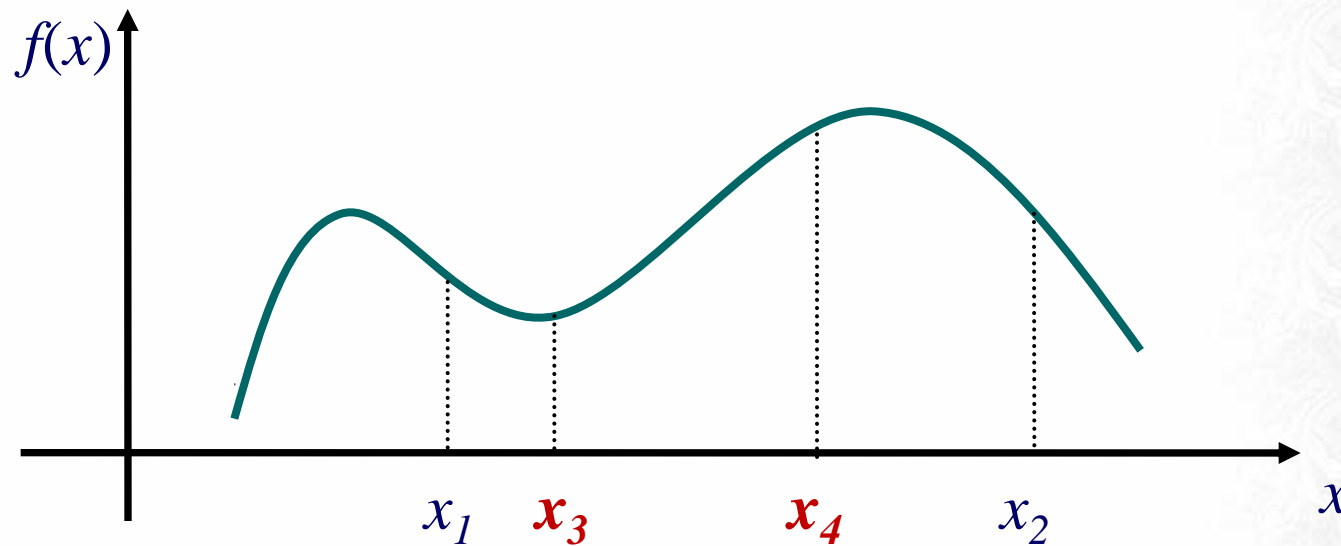
- Por ejemplo: Optimización de una función multimodal.



- La idea consiste en impedir que el individuo x_1 del nicho 1 y se cruce con el individuo x_2 del nicho 2.
 - Si x_1 se cruza con x_2 dominarán las características de x_2 , y no se muestrearán adecuadamente el nicho 1.

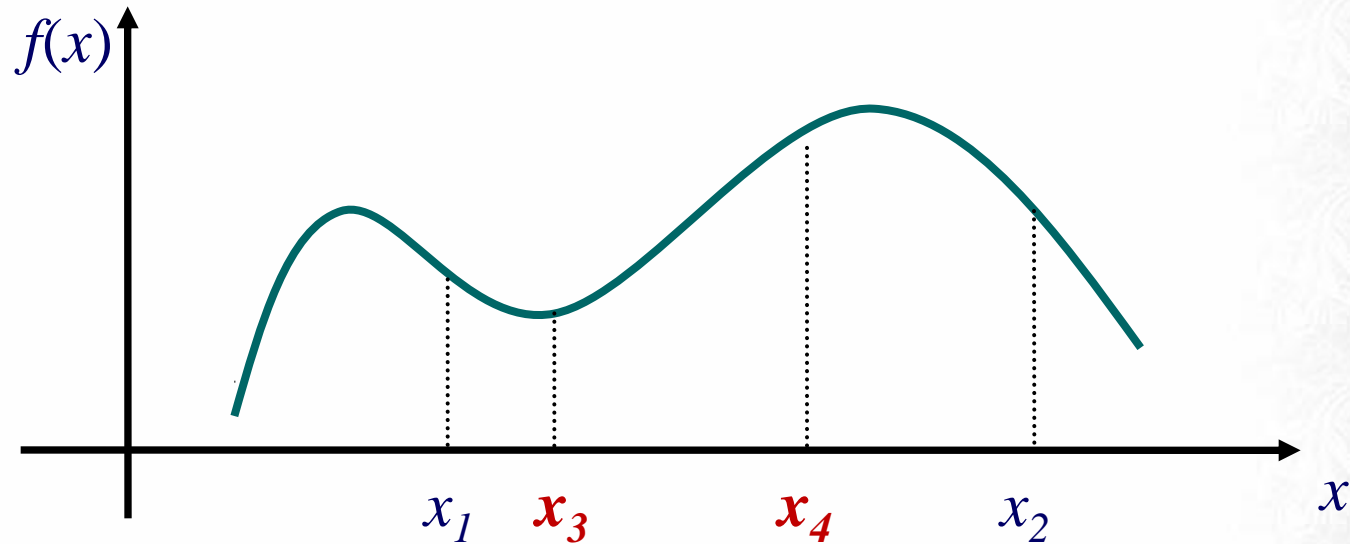
Crowding

- Es una variante propuesta en la que no se incorpora una restricción explícita al cruzamiento.
- En cambio, cada nuevo individuo reemplazará al individuo *más similar* a si mismo existente en la población.



x_1 se cruza con x_2 generando x_3 y x_4 .

Crowding



- x_4 sustituirá a x_2 y preservará a x_1 .
- El parámetro llamado *crowding factor* que determina la cantidad de individuos contra los que se compara cada nuevo individuo.
- Este mecanismo ayuda a mantener la diversidad y *reserva* espacio para nuevas especies.

Crowding determinista

- Cada descendiente (h_1 y h_2) compete en torneo con sus padres (p_1 y p_2), sustituyendo al padre *más cercano* de acuerdo a una función de distancia.

si $[d(p_1, h_1) + d(p_2, h_2)] \leq [d(p_1, h_2) + d(p_2, h_1)]$

si $f(h_1) > f(p_1)$ se sustituye p_1 por h_1

si $f(h_2) > f(p_2)$ se sustituye p_2 por h_2

sino

si $f(h_1) > f(p_2)$ se sustituye p_2 por h_1

si $f(h_2) > f(p_1)$ se sustituye p_1 por h_2

fin

Algoritmo de reemplazo en la técnica de crowding determinista

Fitness sharing

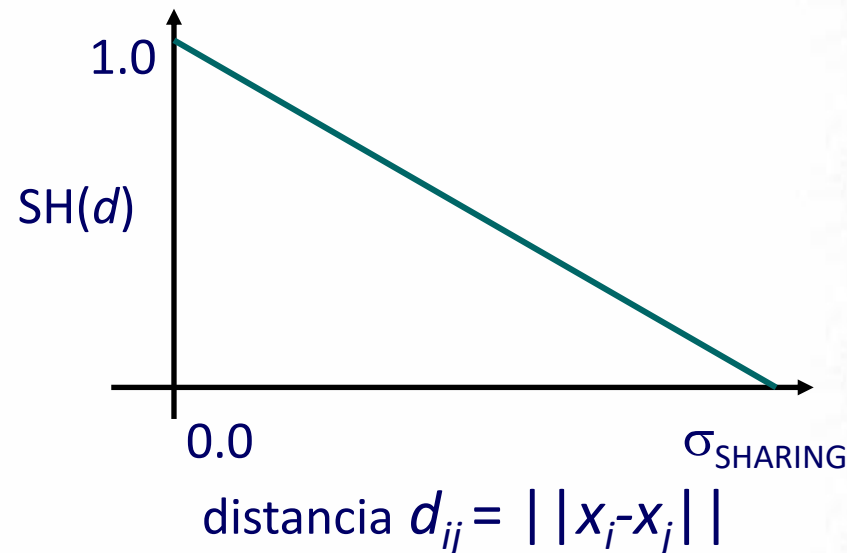
- La idea consiste en decrementar el fitness de las zonas muy representadas del espacio de búsqueda.
- Se determina un *grado de vecindad* entre individuos sumando para cada uno de ellos los valores de una función S (función de Sharing) correspondientes a cada *vecino*.
- Se trabaja con valores de fitness reducido f_R :

$$f_R(x_i) = \frac{f(x_i)}{\sum_j SH(d(x_i, x_j))}$$

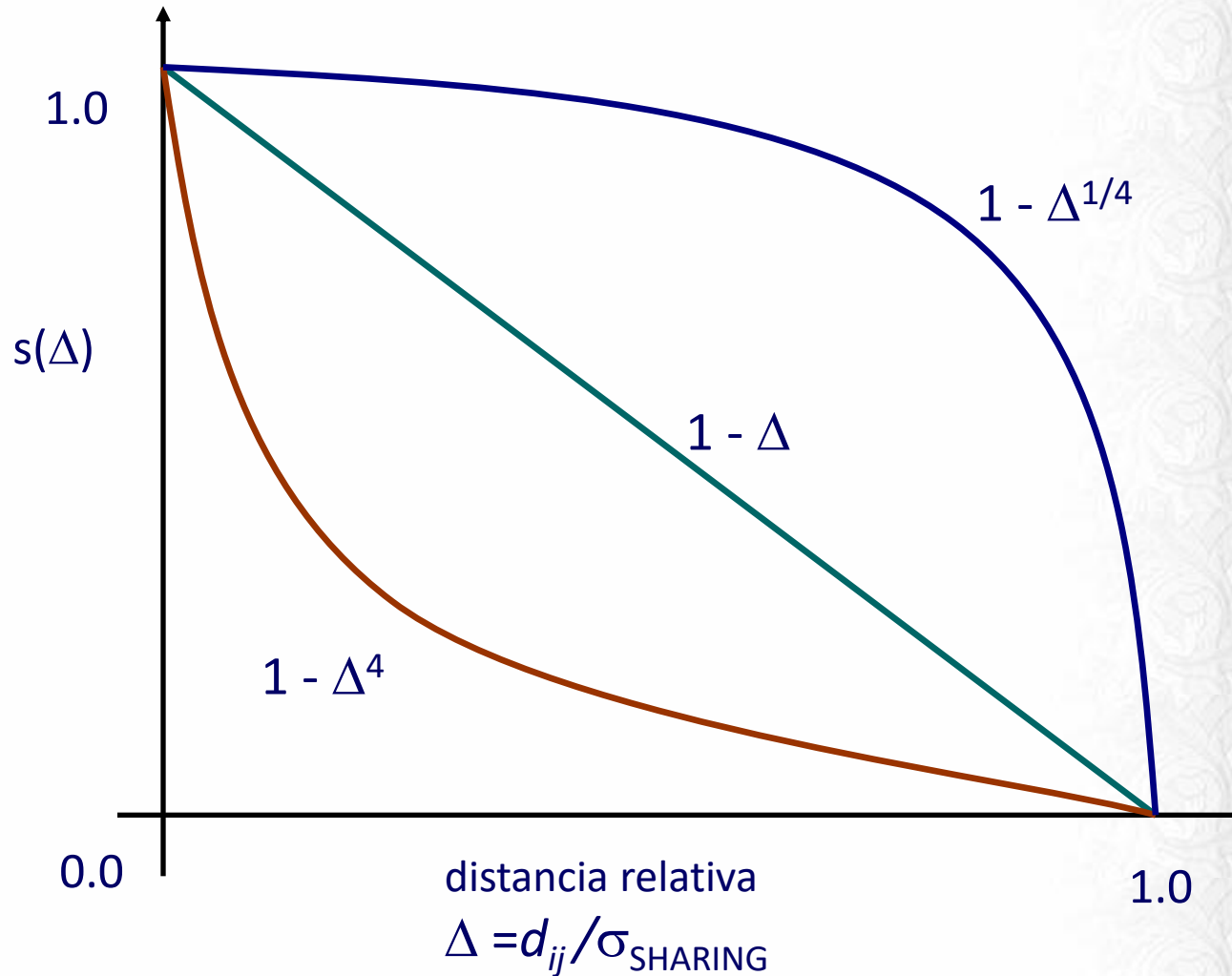
Fitness sharing

- La función de sharing toma valores cercanos a 1 para individuos vecinos o cercanos:

$$SH(d(x_i, x_j)) = \begin{cases} 1 - \left(\frac{d(x_i, x_j)}{\sigma_{\text{SHARING}}} \right) & \text{si } d(x_i, x_j) \leq \sigma_{\text{SHARING}} \\ 0 & \text{si } d(x_i, x_j) > \sigma_{\text{SHARING}} \end{cases}$$



Fitness sharing



Sharing en función de la distancia relativa para sharing potencia



Restricciones al cruzamiento

- Previene (o minimiza) la aparición de individuos de bajo desempeño.
- Para lograrlo se introduce un sesgo en la recombinación de individuos, tratando de incrementar la efectividad y la eficiencia del algoritmo.
- Se presentarán tres propuestas:
 - Castas con cruces ocasionales.
 - Patrones de cruzamiento.
 - Restricción por distancia.

Restricciones al cruzamiento

Castas con cruces ocasionales

- Los cruces se realizan entre *familias* o *castas* que son definidas por características comunes.
- El cruzamiento se mantiene entre individuos de la misma casta mientras el fitness observado mejora.
- Solamente cuando no existe mejora se autoriza el cruzamiento entre individuos de castas diferentes.
- La medida de «mejora» a considerar puede ser el fitness promedio, el mejor fitness, la desviación estándar, o combinaciones de ellas.

Patrones de cruzamiento (“mating templates”)

- Los individuos codifican su valor funcional y un patrón para el apareamiento:

`<temp> : <func>`

`#10# : 1010`

`#01# : 1100`

`#00# : 0000`

- Los # representan posiciones que pueden coincidir con 0 y 1.
- El mapeo de templates para autorizar el cruzamiento puede ser bidireccional, unidireccional o mejor parcial.

Restricción basada en distancia

- El cruzamiento solo está autorizado entre individuos cuyas características difieren como mínimo en un valor dado
 - Se considera una función de distancia.
- El objetivo es intentar lograr un cruzamiento “efectivo” que potencie la exploración del espacio de soluciones.
- Eshelman introdujo esta técnica para mantener la diversidad en su propuesta de algoritmo CHC.