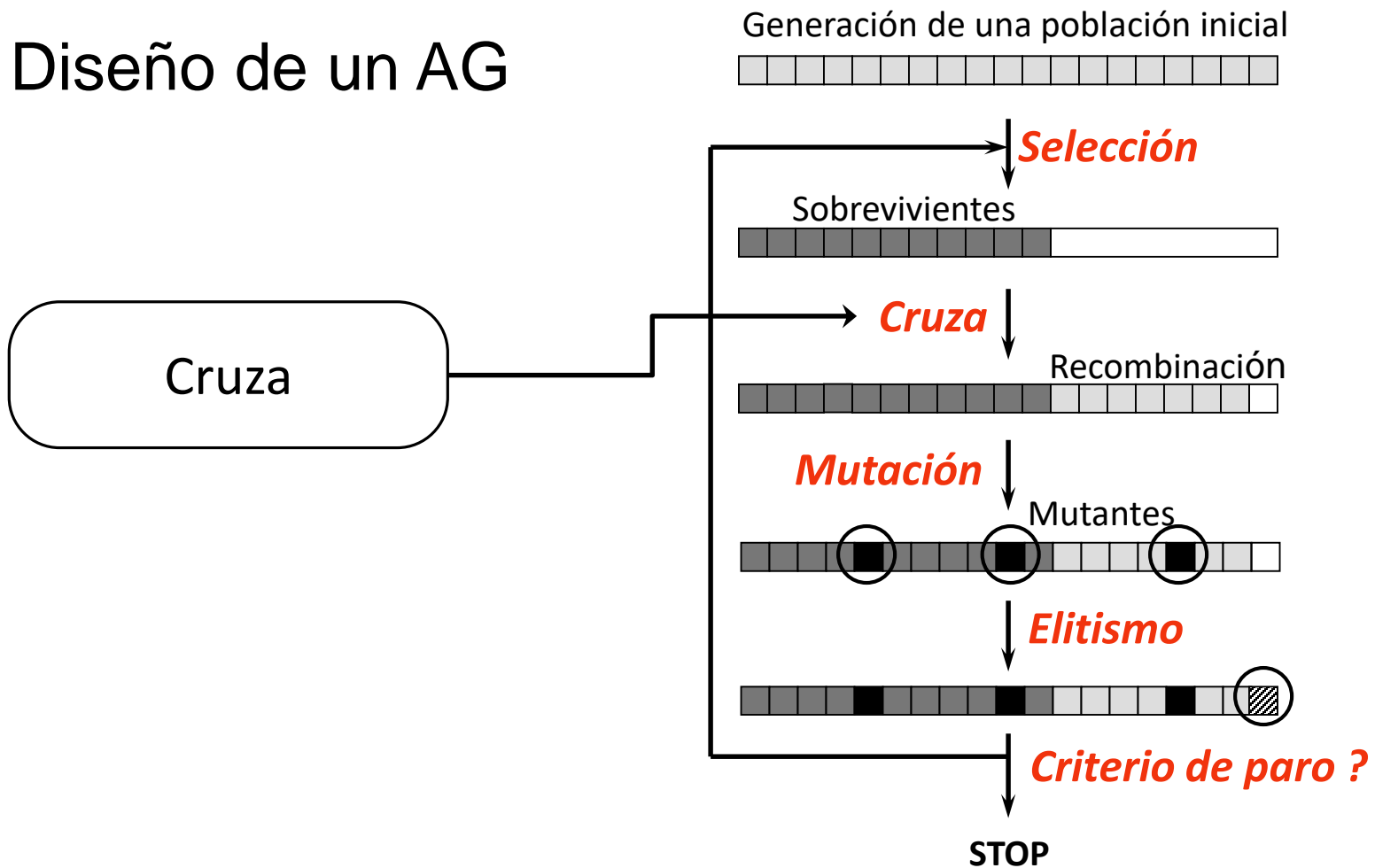


# AGs: Técnicas de Cruza

---

# Diseño de un AG



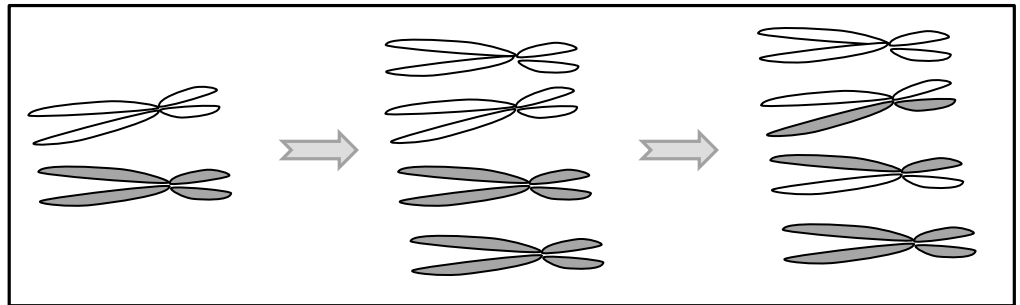
# Introducción

---

## ● Sistemas biológicos

- ▶ Cruza = proceso complejo que ocurre entre parejas de cromosomas

*Meiosis o  
reproducción celular sexual*



## ● Computación evolutiva

- ▶ Cruza simulada mediante el intercambio de segmentos de cadenas lineales de longitud fija (= cromosomas)
- ▶ Técnicas de cruce básicas normalmente aplicadas a representación binaria
- ▶ Desarrollo de esquemas generalizables a alfabetos de cardinalidad mayor

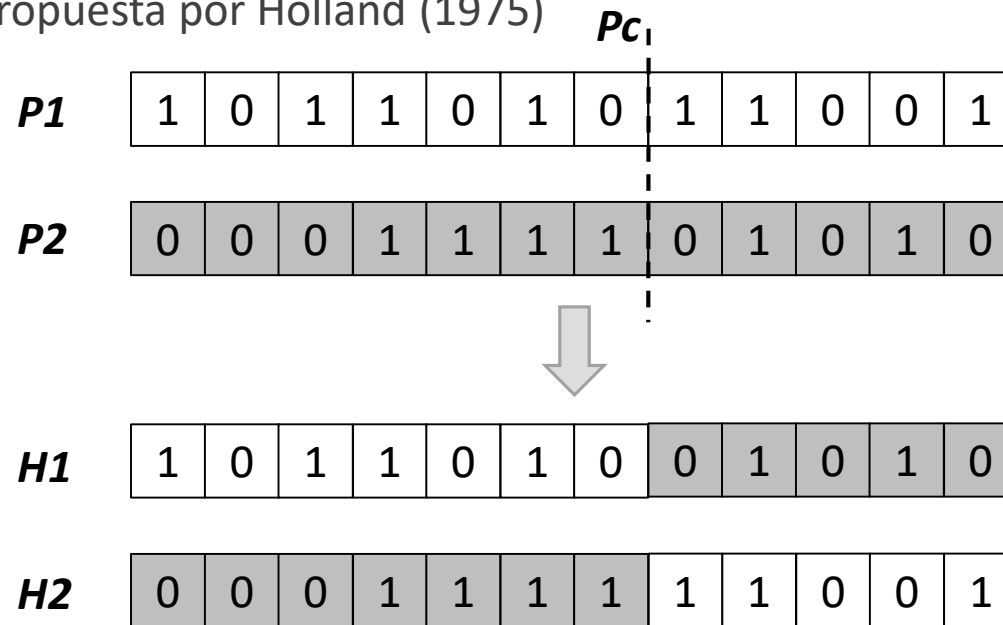
# Cruza de punto

---

# Técnicas clásicas : representación binaria

## ● Cruza de un punto

► Técnica propuesta por Holland (1975)



► Punto de cruza  $P_c$  elegido aleatoriamente

# Técnicas clásicas: representación binaria

---

## Cruza de un punto

- ▶ Muy popular en las primeras décadas de los AGs
- ▶ Menos usada hoy en día debido a su incapacidad a formar ciertos esquemas
- ▶ Ejemplo

$$E_1 = 11*****1$$

$$E_2 = *****11**$$

Si aplicamos cruza de un punto a estos 2 esquemas, no hay manera de formar una instancia del esquema:

$$H = 11**11*1$$

# Técnicas clásicas: representación binaria

---

## Definiciones

- Orden  $O$  de un esquema  $E$

$O(E)$  = número de posiciones fijas en un esquema

$$E = 01^{**}10^{*}1 \quad \Rightarrow \quad O(E) = 5$$

- Longitud de definición  $\delta$  de un esquema  $E$

$\delta(E)$  = distancia entre la primera y la última posición fija de un esquema

$$E = 01^{**}10^{*}1 \quad \Rightarrow \quad \delta(E) = 7$$

$$E = *1^{****} \quad \Rightarrow \quad \delta(E) = 0$$

Aplicaciones reales suelen requerir cadenas largas: cruza de un punto no adecuada

Cruza de dos  
puntos

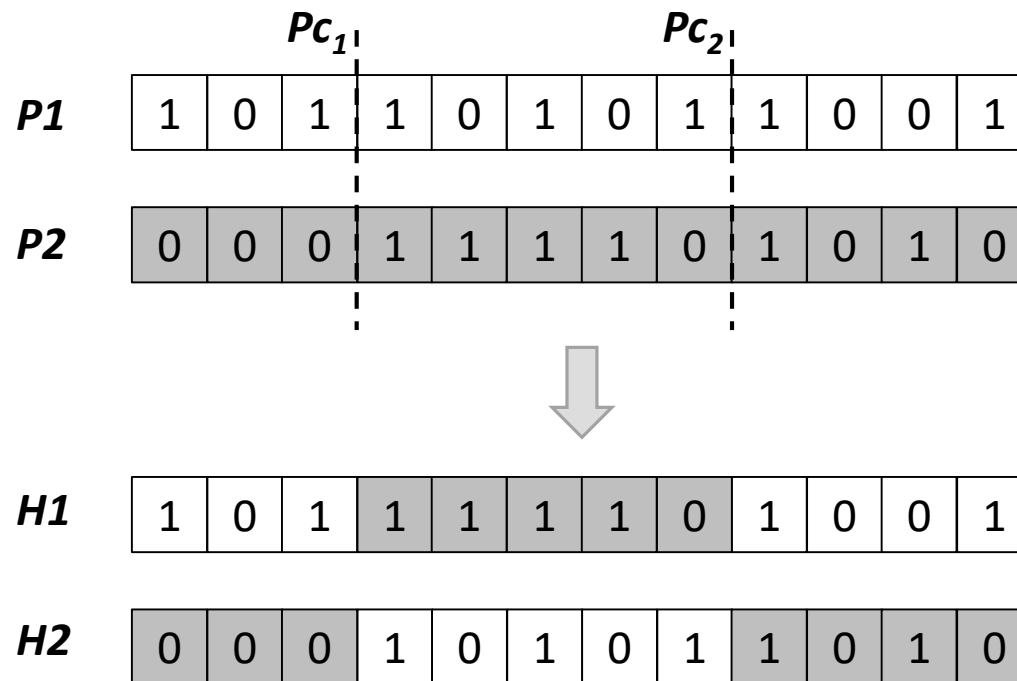
---



# Técnicas clásicas: representación binaria

## Cruza de dos puntos

- Propuesta por De Jong (1975): generalización de la cruce de un punto



# Técnicas clásicas: representación binaria

---

## Cruza de dos puntos

- ▶  $n = 2$  puntos de cruce: valor que minimiza los efectos disruptivos (o destructivos) de la cruce. Técnica muy utilizada.
- ▶ Para  $n > 2$ , no existe consenso sobre valor más adecuado (a pesar de varios estudios empíricos)

## Generalmente se considera que

- ▶ La cruce de dos puntos es mejor que la cruce de un punto
- ▶ Incrementar el valor de  $n$  implica un mayor efecto disruptivo de la cruce

# Cruza uniforme

---

# Técnicas clásicas: representación binaria

---

## Cruza uniforme

- Técnica fue propuesta originalmente por Ackley (1987), aunque a veces atribuida a Syswerda (1989)

**P1**

1	0	1	1	0	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---

**P2**

0	0	0	1	1	1	1	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---



**H1**

1	0	0	1	1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---

**H2**

0	0	1	1	0	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

# Técnicas clásicas: representación binaria

---

## Cruza uniforme

- ▶ Equivalente a una cruce de  $n$  puntos, donde  $n$  no se fija previamente
- ▶ Mayor efecto disruptivo que cualquiera de las 2 cruces anteriores.
- ▶ Suele usarse una probabilidad de cruce  $P_c = 0.5$ 
  - Algunos investigadores, sin embargo, sugieren usar valores más pequeños (para disminuir el efecto disruptivo)

# Cruza acentuada

---

# Técnicas clásicas: representación binaria

---

## Cruza acentuada

- ▶ Técnica propuesta por Schaffer y Morishima (1987), sugerida por Holland (1975)
- ▶ Objetivo: desarrollo de un mecanismo de auto-adaptación para la generación de patrones favorables (bloques constructores) en la cruce

# Técnicas clásicas: representación binaria

---

## Metodología de la cruza acentuada

- ▶ Usa una cadena binaria de “marcas” indicando la localización de los puntos de cruza
- ▶ Información extra (de la cruza acentuada) agregada al cromosoma
  - El número y localizaciones de los puntos de cruza puede ser objeto de manipulación por el AG
  - Las cadenas tendrán por tanto una longitud del doble de su tamaño original



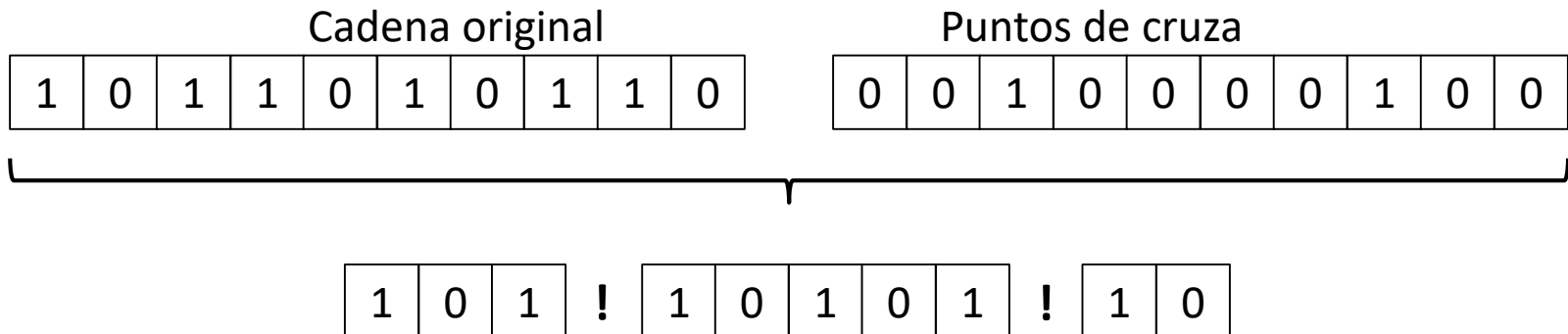
# Técnicas clásicas: representación binaria

---

## Cruza acentuada

### ► Convención clásica

- Marcar con '1' las posiciones donde hay cruce y con '0' las posiciones donde no la hay
- Usar signos de admiración para facilitar la ubicación de los puntos de cruce



# Técnicas clásicas: representación binaria

---

## Algoritmo de la cruza acentuada

- Copiar los bits de cada padre hacia sus hijos, de uno en uno
- Cuando se encuentra un signo de admiración (en cualquiera de los padres), se efectúa la cruza (es decir, se invierte la procedencia de los bits en los hijos)
- Cuando esto ocurre, los signos de admiración se copian también a los hijos, justo antes de que la cruza se efectúe

**P1** = a a a a a a a! b b b b b b b

**P2** = c c c c! d d d d d d! e e e e



**H1** = a a a a! d d d! b b b! e e e e

**H2** = c c c c! a a a! d d d! b b b b

# Técnicas clásicas: representación binaria

---

## Análisis de la cruza acentuada

- ▶ Se usa sólo la primera parte de la cadena para calcular la aptitud
- ▶ Se espera que los operadores genéticos tengan un efecto positivo sobre los puntos de cruza
- ▶ La mutación actúa sobre los dos segmentos cromosómicos
- ▶ Inspiración biológica
  - Marcas de cruza existen en la naturaleza y co-evolucionan junto con los cromosomas
- ▶ Buenos resultados reportados (Schaffer & Morishima, 1987) sobre algunas funciones de prueba
  - No hay evidencia contundente acerca de su efectividad

# Comportamiento deseable

---

# Aspectos teóricos

---

## Comportamiento deseable de la cruce

- ▶ Operadores descritos anteriormente siguen el principio Mendeliano de la herencia: cada gene de un hijo es una copia de un gene heredado de alguno de sus padres. Pero esto no necesariamente tiene que ser así en computación evolutiva
- ▶ Énfasis de la cruce: ser capaz generar todas las posibles combinaciones de bits (de longitud  $L$ ) que hayan en el espacio de búsqueda del problema

# Aspectos teóricos

---

Comportamiento deseable de la cruce

- ▶ Representación binaria: capacidad de generar todas las combinaciones de bits
  - Cruce de  $n$  puntos ( $n \geq 1$ ) no es capaz de lograrlo
  - Cruce uniforme sí puede hacerlo

De acuerdo a esta observación, otras variantes propuestas

# Forma de apareamiento

---

# Aspectos teóricos

---

Formas de apareamiento. Se decide “quien se puede recombinar con quien”

- ▶ Formas más comunes de apareamiento (según Golberg, 1989)
  - Random Mating (aleatorio). Se eligen los individuos aleatoriamente, con la misma probabilidad
  - Inbreeding (entre parientes). Se recombinan individuos similares
  - Line Breeding (semental). Un solo super-individuo (aptitud alta) se recombina con una población base y sus hijos se seleccionan como padres



# Aspectos teóricos

---

## Formas de apareamiento

- ▶ Formas más comunes de apareamiento (según Golberg, 1989)
  - Outbreeding (entre desconocidos). Sólo se recombinan individuos muy diferentes
  - Self-fertilization (auto-fertilización). Un individuo se recombina con sí mismo.
  - Cloning (clonación). Un individuo se copia sin modificaciones

# Aspectos teóricos

---

## Recombinación con padres múltiples

- ▶ Algunos desarrollos aunque esta técnica no es muy común
- ▶ Diagonal crossover (Eiben et al., 1995)
  - Creación de  $p$  hijos a partir de  $p$  padres
  - Se elige un número de puntos de cruza  $N \geq 1$
  - El primer hijo contiene el  $i$ -ésimo segmento del padre  $i$ ,  $i=1, \dots, p$
  - Otros hijos: contruidos de una rotación de segmentos de los padres

# Cruza para permutaciones

---

# Cruza para permutaciones

---

## Generalidades

- ▶ Permutaciones frecuentemente usadas en problemas de optimización combinatoria (viajero, scheduling, etc.)
- ▶ Cruzas antes descritas entre cadenas de permutaciones produce invariablemente hijos no válidos
  - ▶ Procedimientos de reparación de las cadenas inválidas
  - ▶ Cruzas adaptadas

# Cruza para permutaciones

---

## Order Crossover (OX)

- ▶ Técnica propuesta por Davis (1985)
- ▶ Algoritmo (con padres P1 y P2)
  - Seleccionar (aleatoriamente) una sub-cadena de P1
  - Producir un hijo copiando la sub-cadena en las posiciones correspondientes a P1. Las posiciones restantes se dejan en blanco
  - Borrar los valores que ya se encuentren en la sub-cadena de P2. La secuencia resultante contiene los valores faltantes
  - Colocar los valores en posiciones no conocidas del hijo de izquierda a derecha
  - Para obtener el segundo hijo, se repiten los pasos del 1 al 4, pero tomando ahora la sub-cadena de P2

# Cruza para permutaciones

---

Order Crossover: ejemplo

P1 = 9 8 4 5 6 7 1 2 3 10

P2 = 8 7 1 2 3 10 9 5 4 6

Sub-cadena elegida: 5 6 7 1 (de P1)

Primer hijo: H1 = X X X 5 6 7 1 X X X

Borrar de P2 la sub-cadena tomada de P1:

P2' = 8 X X 2 3 10 9 X 4 X

Finalmente: H1 = 8 2 3 5 6 7 1 10 9 4

Para H2: procedimiento similar tomando la sub-cadena de P2 y sustituyendo partir de P1'

# Cruza para permutaciones

---

## Partially Mapped Crossover (PMX)

► Técnica propuesta por Goldberg y Lingle (1985), tiene ciertas similitudes con OX

► Algoritmo

- Elegir aleatoriamente dos puntos de cruce

- Intercambiar segmentos centrales como la cruce de 2 puntos convencional

- Resto de las cadenas obtenido realizando mapeos entre los 2 padres

- a) Si un valor no está contenido en el segmento intercambiado, permanece igual
- b) Si está contenido en el segmento intercambiado, entonces se sustituye por el valor que tenga dicho segmento en el otro padre, de derecha a izquierda

# Cruza para permutaciones

---

Partially Mapped Crossover: ejemplo

P1 = 9 8 4 | 5 6 7 | 1 3 2 10

P2 = 8 7 1 | 2 3 10 | 9 5 4 6

Hijos H1 = X X X | 2 3 10 | X X X X

H2 = X X X | 5 6 7 | X X X X

Copiar primero los valores que no están en el segmento intercambiado

H1 = 9 8 4 | 2 3 10 | 1 X X X

H2 = 8 X 1 | 5 6 7 | 9 X 4 X

Mapeamos los valores restantes

H1 = 9 8 4 2 3 10 1 6 5 7 y H2 = 8 10 1 5 6 7 9 2 4 3



# Cruza para permutaciones

---

## ● Order-based Crossover: ejemplo

P1 = 9 8 4 5 6 7 1 2 3 10

P2 = 8 7 1 2 3 10 9 5 4 6

Valores elegidos de P1: 8 6 2 10

Remover estos valores de P2: P2' = X 7 1 X 3 X 9 5 4 X

Producir un hijo: H1 = X 7 1 X 3 X 9 5 4 X

Insertar ahora la secuencia elegida de P1:

H1 = 8 7 1 6 3 2 9 5 4 10