

Variaciones en un Algoritmo Genético

Héctor Alejandro Montes

hamontesv@uaemex.mx

<http://scfi.uaemex.mx/hamontes>

Advertencia

No use estas diapositivas como referencia única de estudio durante este curso. La información contenida aquí es sólo una guía para las sesiones de clase y de estudio futuro. Para obtener información más completa, refiérase a la bibliografía dada durante la presentación del curso.

Pasos para construir un AG

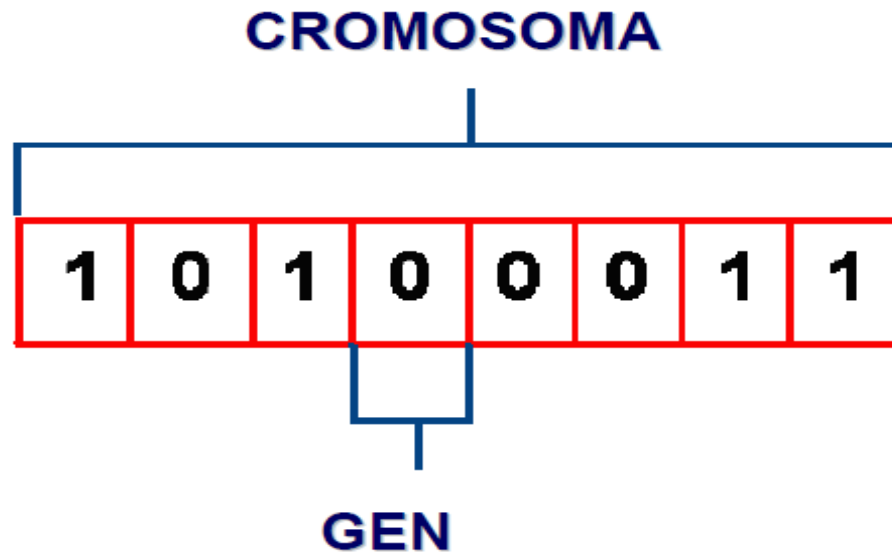
- Dependientes del problema
 - Diseñar una *representación*
 - Decidir cómo *inicializar* la población
 - Diseñar una forma de *evaluar* a los individuos
- Inherentes al AG
 - Decidir cómo *seleccionar* los individuos para ser padres
 - Diseñar un operador de *cruza* adecuado
 - Diseñar un operador de *mutación* adecuado
 - Decidir cómo *reemplazar* a los individuos
- Decidir la condición de paro

Representación

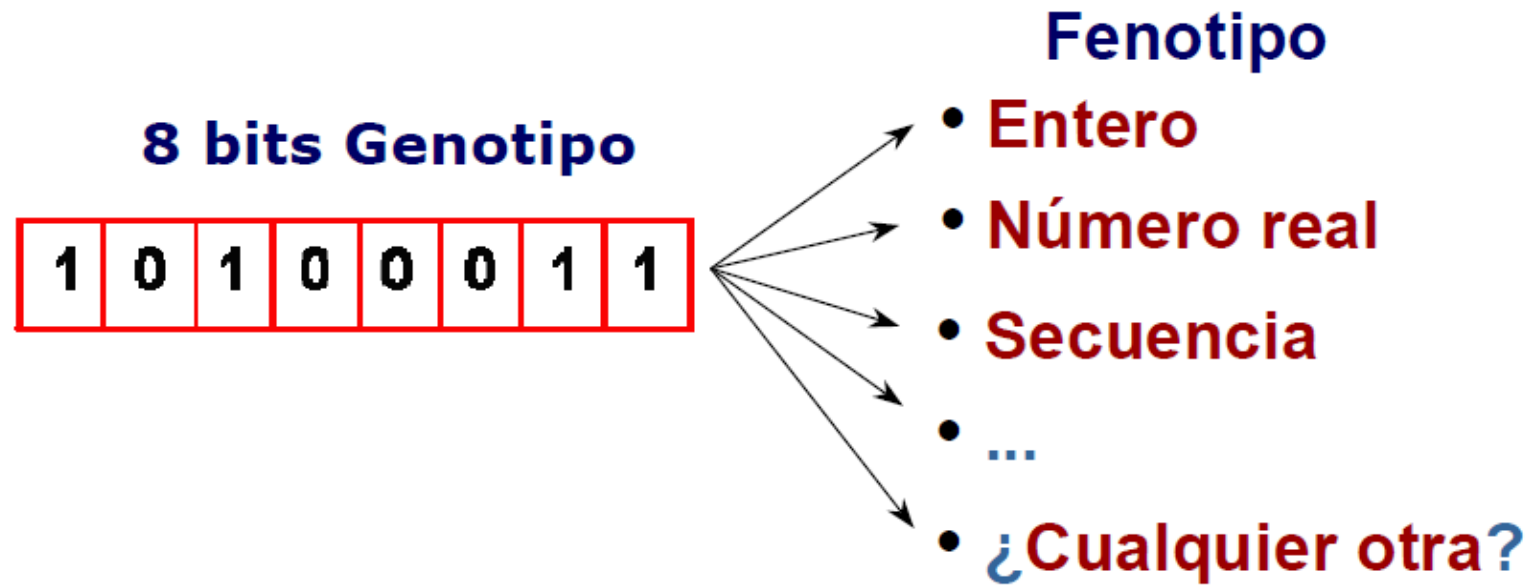
- Debemos disponer de un *mecanismo para codificar* un individuo como un **genotipo**
- Existen muchas maneras de hacer esto y se debe elegir la más relevante (*¿adecuada?*) para el problema a tratar
- Una vez elegida una representación, tenemos que tener en mente cómo se evaluarán los *genotipos* (codificación) y qué *operadores genéticos* habrá que utilizar

Representación binaria

- La representación de un individuo se puede hacer mediante una **codificación discreta**, en particular **binaria**



Representación binaria




Representación binaria

Genotipo:

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

Fenotipo:

= 163


$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 128 + 32 + 2 + 1 = 163$$

Representación binaria


- También puede corresponder a **números reales**.
Ejemplo: un número entre 2.5 y 20.5 utilizando 8 dígitos binarios

Genotipo:

1 0 1 0 0 0 1 1

Fenotipo:

= 13,9609


$$x = 2,5 + \frac{163}{256} (20,5 - 2,5) = 13,9609$$

Representación Real

- Una forma natural de codificar una solución es utilizando valores reales como genes
- Muchas aplicaciones tienen esta forma natural de codificación

Representación Real

- Los individuos se representan como vectores de valores reales:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, x_i \in R$$

- La función de evaluación asocia a un vector un valor real de evaluación:

$$f : R^n \rightarrow R$$

Representación de Permutación

- Conocida también como de *Orden* o *Secuencia*
- Se utilizan para problemas de secuenciación
- Ejemplo: *El Agente Viajero* (TSP), donde cada ciudad tiene asignado un único número entre *1* y *n* (o cualquier otra etiqueta)
- Necesita **operadores especiales** para garantizar que el resultado de aplicar un operador sea una **permutación válida**

Inicialización

- Aleatoria (**uniforme**) sobre el espacio de búsqueda
 - Representación **binaria**: 0 o 1 con probabilidad 0.5
 - Representación **real**: uniforme sobre un intervalo dado (para valores acotados)
- Elegir la población a partir de los resultados de una heurística previa

Evaluación (fitness)

- Es el paso *más costoso* para una aplicación real
- Puede ser una subrutina, un simulador, o cualquier proceso externo (ejem., experimentos en un robot, procesamiento de imágenes, visión artificial, etc.)
- Se pueden utilizar *funciones aproximadas* para reducir el costo de evaluación
- Cuando hay *restricciones*, éstas se pueden introducir en la función de fitness como *penalización* del costo
- Con *múltiples objetivos* se busca una solución de compromiso

Selección

- Debemos de garantizar que los mejores individuos tienen una *mayor probabilidad* de ser padres (reproducirse).
- Aunque los individuos *menos aptos*, también deben tener oportunidad de reproducirse (pueden aportar material útil en el proceso de reproducción).
- Este proceso define la *presión selectiva* que determina en qué grado la reproducción está dirigida por los mejores individuos.

Esquemas de Selección

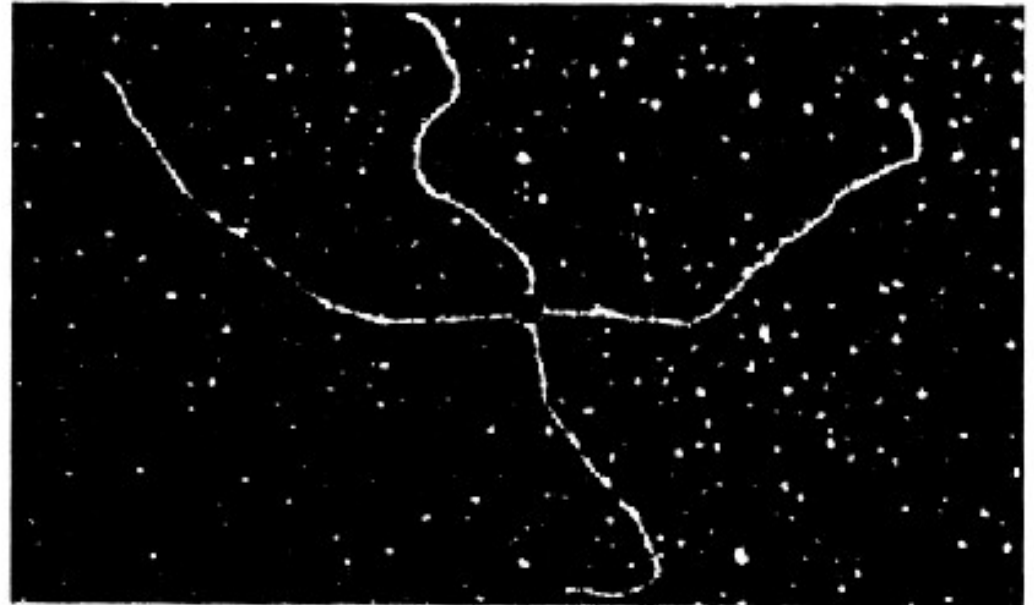
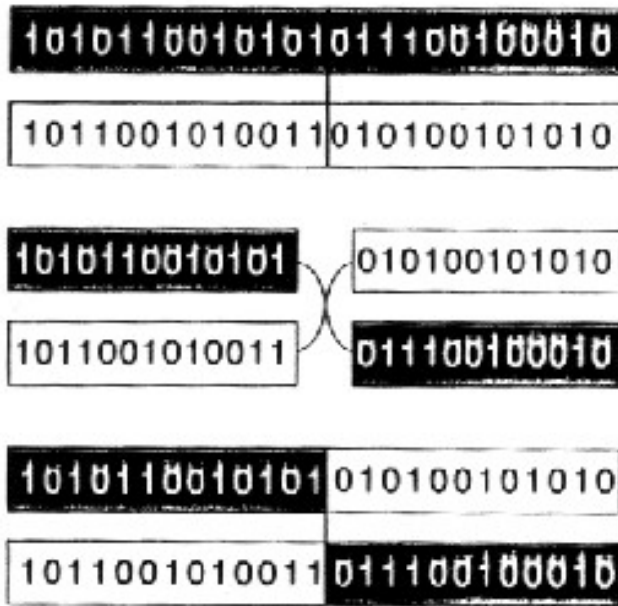
- **Selección por Ruleta (modelo clásico):** Se asigna una probabilidad de selección proporcional al valor del fitness del cromosoma.
- **Selección por Torneo (TS):** Escoge al individuo de mejor fitness de entre N individuos seleccionados aleatoriamente ($N = 2, 3, \dots$).
- **Orden Lineal (o de rango, LR):** La población se ordena en función de su *fitness* y se asocia una probabilidad de selección a cada individuo que depende de su orden.

Cruza (crossover)

- Podríamos tener *uno o más* operadores de cruce para nuestra representación.
- Aspectos importantes a tomar en cuenta incluyen:
 - Los hijos deberían heredar algunas características de *cada padre*. Si éste no es el caso, entonces estamos ante un operador de *mutación*.
 - Se debe diseñar de acuerdo a la representación.
 - La recombinación debe producir cromosomas *válidos*.
 - Se utiliza con una probabilidad alta sobre los padres a cruzar (**PC** entre **0.6** y **0.9**).
 - Si *no* se efectúa, los mismos padres son los *descendientes directos*.

Cruza (crossover)

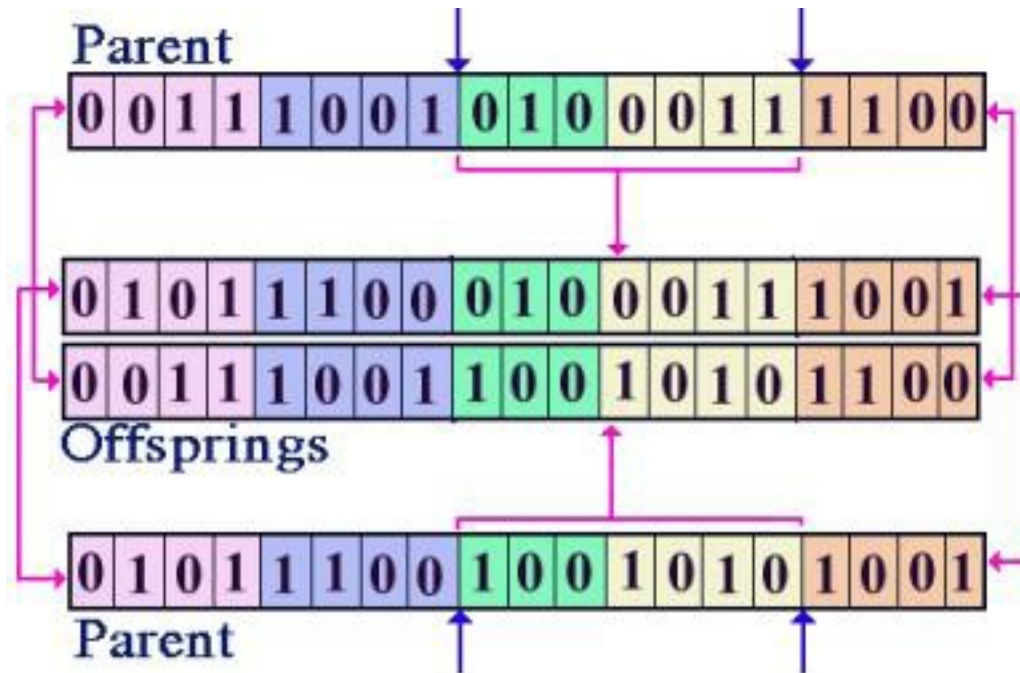
- Imagen original de *John Holland* donde se introduce el operador de cruza



CROSSOVER is the fundamental mechanism of genetic rearrangement for both real organisms and genetic algorithms.

Chromosomes line up and then swap the portions of their genetic code beyond the crossover point.

Cruza de dos puntos



Cruza para representación Real

- Recombinación discreta (cruce uniforme): dados 2 padres se crea un *descendiente*:

a	b	c	d	e	f	g	h
---	---	---	---	---	---	---	---



A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---

a	b	C	d	E	f	g	H
---	---	---	---	---	---	---	---

- Existe muchos *operadores específicos* para la codificación real.

Cruza para representación Real

- Recombinación aritmética (cruce aritmético):

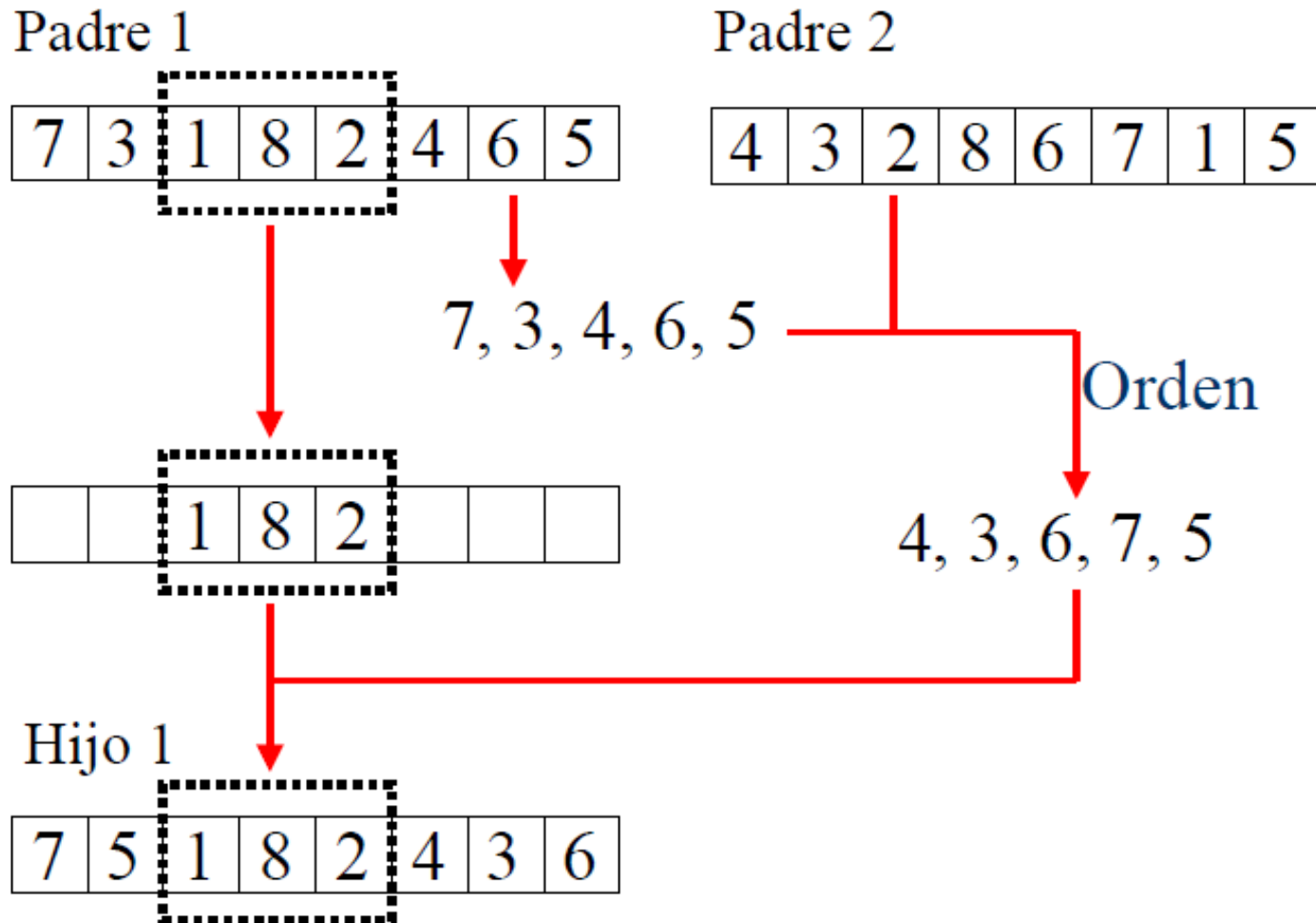
a	b	c	d	e	f
---	---	---	---	---	---

A	B	C	D	E	F
---	---	---	---	---	---



$(a+A)/2$	$(b+B)/2$	$(c+C)/2$	$(d+D)/2$	$(e+E)/2$	$(f+F)/2$
-----------	-----------	-----------	-----------	-----------	-----------

Cruza OX para Permutaciones

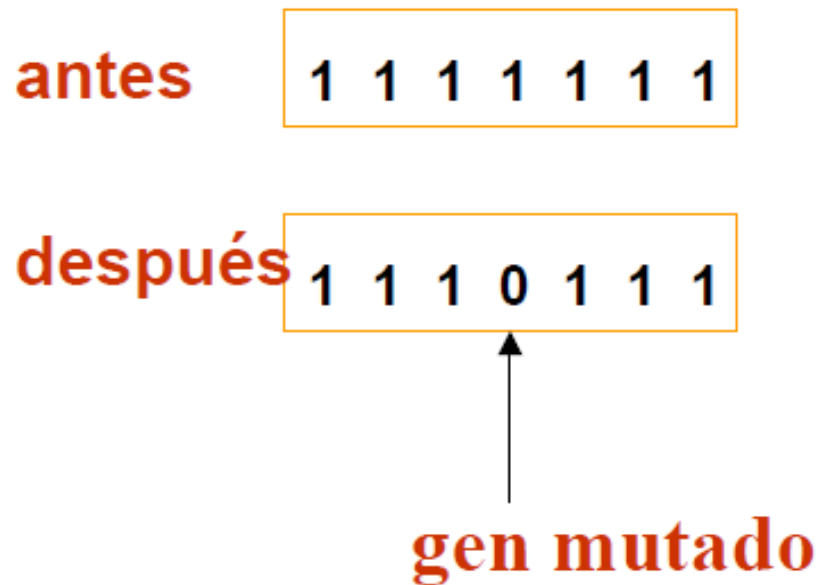


Mutación

- Podemos tener *uno o más* operadores de mutación para nuestra representación.
 - Debe permitir *alcanzar cualquier* parte del espacio de búsqueda.
 - El *tamaño* de la mutación debe ser controlado.
 - Debe producir cromosomas *válidos*.
 - Se aplica con una probabilidad *muy baja* (0.03) y usualmente después de aplicar el operador de cruce

Mutación para representación Binaria

- La mutación ocurre con una probabilidad ***PM*** para cada gen

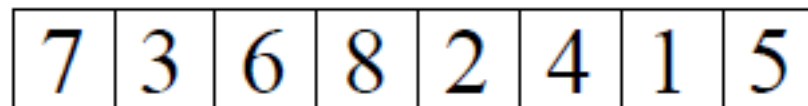
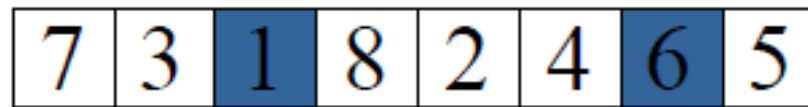


Mutación para representación Real

- Perturbación de los valores mediante un valor aleatorio.
 - Frecuentemente, mediante una distribución *Gaussiana/normal* $N(m, \sigma)$, donde
 - m es la media
 - σ es la desviación estándar
- $x'_i = x_i + N(m, \sigma_i)$ para cada parámetro (gen) x_i

Mutación para representación de Permutación

- Selección aleatoria de dos genes e intercambio de ambos.



Reemplazo

- La *presión selectiva* se ve también afectada por la forma en que los cromosomas de la población son reemplazados por los nuevos descendientes.
- Podemos utilizar métodos de reemplazamiento *aleatorios* o *determinísticos*.
- Podemos decidir **no** reemplazar al mejor cromosoma de la población y simplemente transferirlo: *Elitismo*.
- Se puede también seleccionar los **N** mejores (o usar otra estrategia).

Criterio de paro

- Cuando se alcanza el *óptimo*
- Recursos de cómputo son *limitados*, se pueden fijar un número máximo de evaluaciones
- Decidido directamente por *el usuario*: Después de algunas iteraciones sin mejora
- *Estadísticamente*: ejemplo, cuando la varianza es poca o nula.

Consideraciones importantes

- Nunca sacar conclusiones de una única ejecución
 - Utilizar *medidas estadísticas* (media, mediana, ...)
 - Con un *número suficiente* de ejecuciones independientes
- “He obtenido la solución en una sóla experimentación sobre un problema”
 - **No** se debe asumir la actuación de un algoritmo sobre ejemplos simples si se desea trabajar con casos reales
- Desde el punto de vista de las aplicaciones:
 - Encontrar *una* solución de *buena calidad* al menos una vez
 - Encontrar *al menos* una solución de *buena calidad* en cada ejecución