



ALGORITMOS GENÉTICOS

Laboratorio Inteligencia Artificial



Algoritmos Genéticos

- Los Algoritmos Genéticos son métodos adaptativos que son utilizados para resolver problemas de búsqueda y optimización. Estos algoritmos están basados en el proceso genético de los seres vivos.
- Los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real.
- Se busca que las soluciones dadas por el algoritmo evolucionen hacia valores óptimos del problema atacado.

Elementos dentro de un Algoritmo Genético

- Individuo: es una posible solución al problema. Estos están conformados por un conjunto de parámetros a los cuales vamos a llamar genes. Estos individuos van a estar representados como un arreglo dentro de nuestro código (un arreglo de genes).
- Población: es un conjunto de soluciones (individuos).

¿Cuál es el principal objetivo de los algoritmos genéticos?

OPTIMIZAR

EJEMPLO

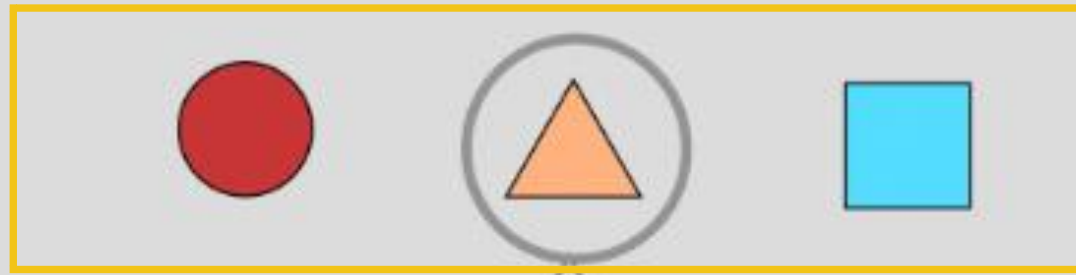
■ Problema



Quiero conseguir una estrella naranja

Población

Individuo



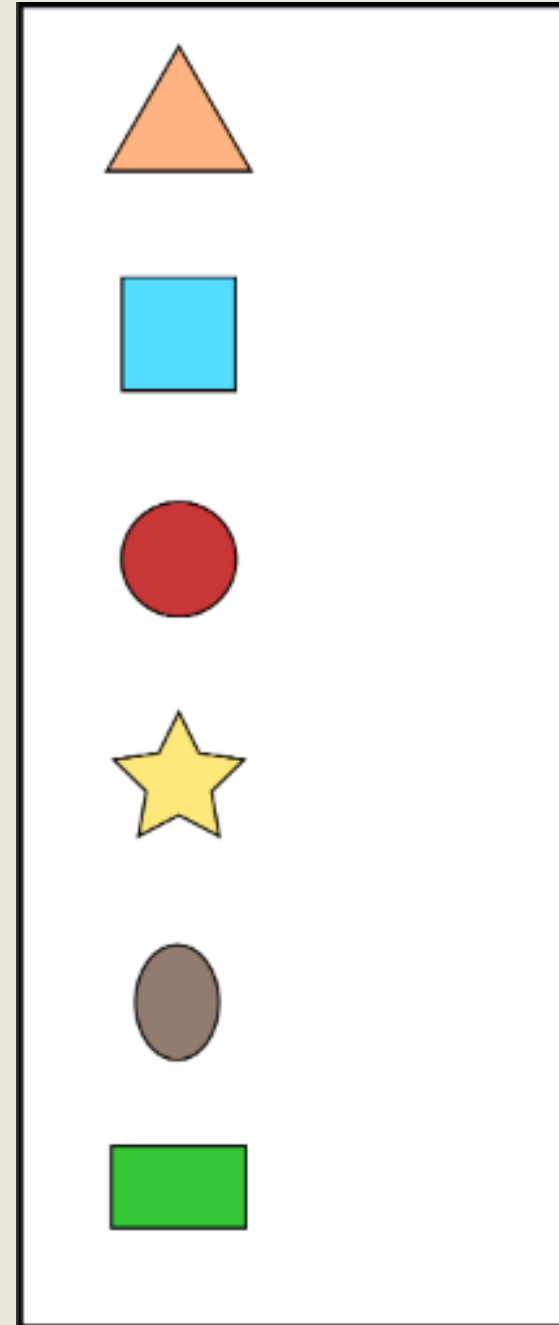
Secuencia de un algoritmo genético

- Básicamente es un proceso iterativo que se realiza sobre una población de individuos. Se puede resumir en los siguientes pasos:
 - Inicialización de la población
 - Evaluación de un individuo
 - Selección de padres (se toman solo ciertos padres de una generación)
 - Emparejamiento (cómo se van a emparejar los padres para crear nuevos hijos)
 - Cruzar (cómo se van a conformar los nuevos hijos, qué elementos de cada padre se van a utilizar)
 - Mutación (qué alteraciones va a tener un nuevo hijo)

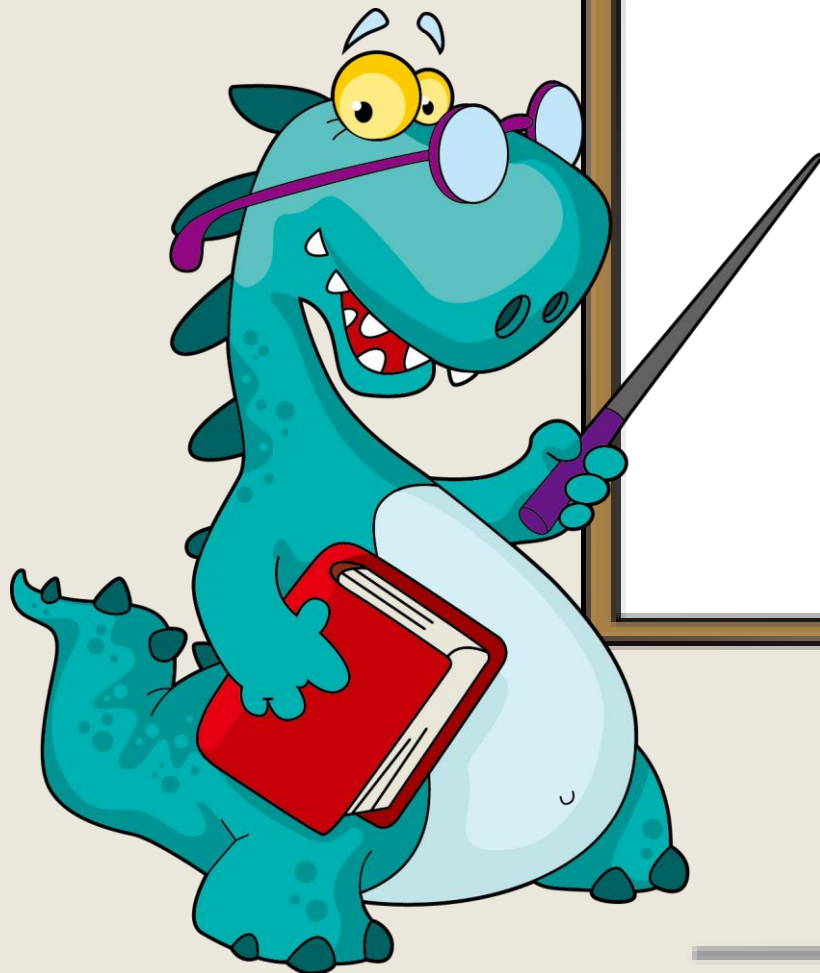
Inicialización

- Se crea la población inicial con individuos al azar.

$$P_0 = [S_1, S_2, \dots, S_N]$$



Consejos



- La población inicial debe seleccionarse al azar.
- El tamaño de la población debe ser lo suficientemente grande para que la variación entre las soluciones sea grande.

Evaluación

- Se evalúa la calidad de todos los individuos de la población, se les asigna un valor numérico llamado *fitness*.

E0 = []

For every S in P0:

temp = fitness(S)

E0.push(temp)



4



1



2



5



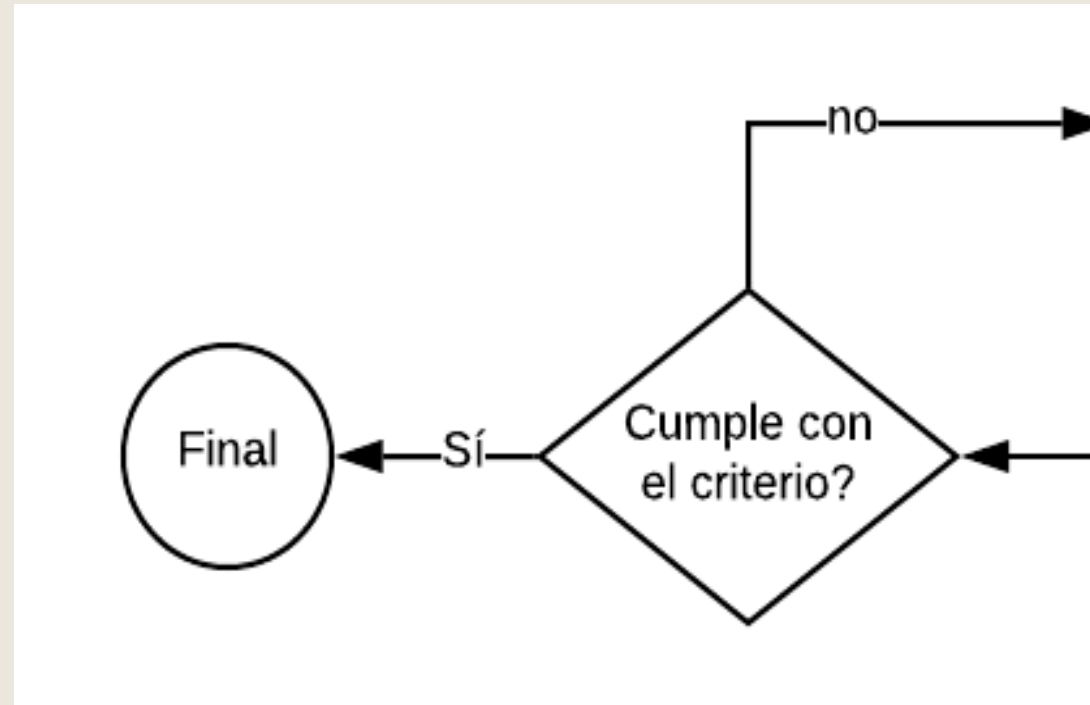
0



1

Fin del algoritmo

- Después de la evaluación, si alguna de las soluciones cumple el criterio de aceptación el algoritmo termina y la mejor solución de la población pasa a ser la solución del algoritmo.









Selección

- Se eligen a los individuos con el valor **fitness** mas alto.

NP = []

NP = max(P0, E0)

	4	➡
	1	
	2	
	5	➡
	0	
	1	

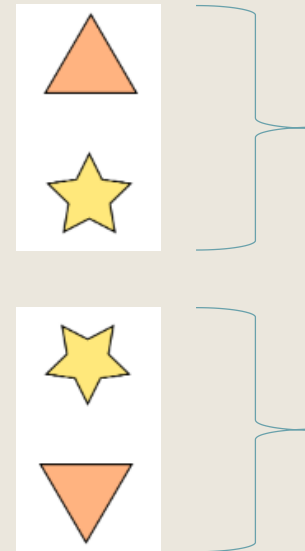
Emparejamiento

- Se emparejan los individuos seleccionados para la operación de reproducción (*Cruzar*).

$NP = [S_0, S_1, S_2, \dots, S_N]$

$Pair1 = \text{random}(NP, 2)$

$Pair2 = \text{random}(NP, 2)$



Cruzar

- Se mezclan las características de los padres seleccionados para generar hijos que sustituyan a los padres que no fueron seleccionados.

Hijo1 = merge(Pair1)

Hijo2 = merge(Pair2)

Hijo3 = merge(Pair3)

Hijo4 = merge(Pair4)

Hijo1 →



Hijo2 →



Padre1 →



Padre2 →



Hijo3 →



Hijo4 →



Mutación

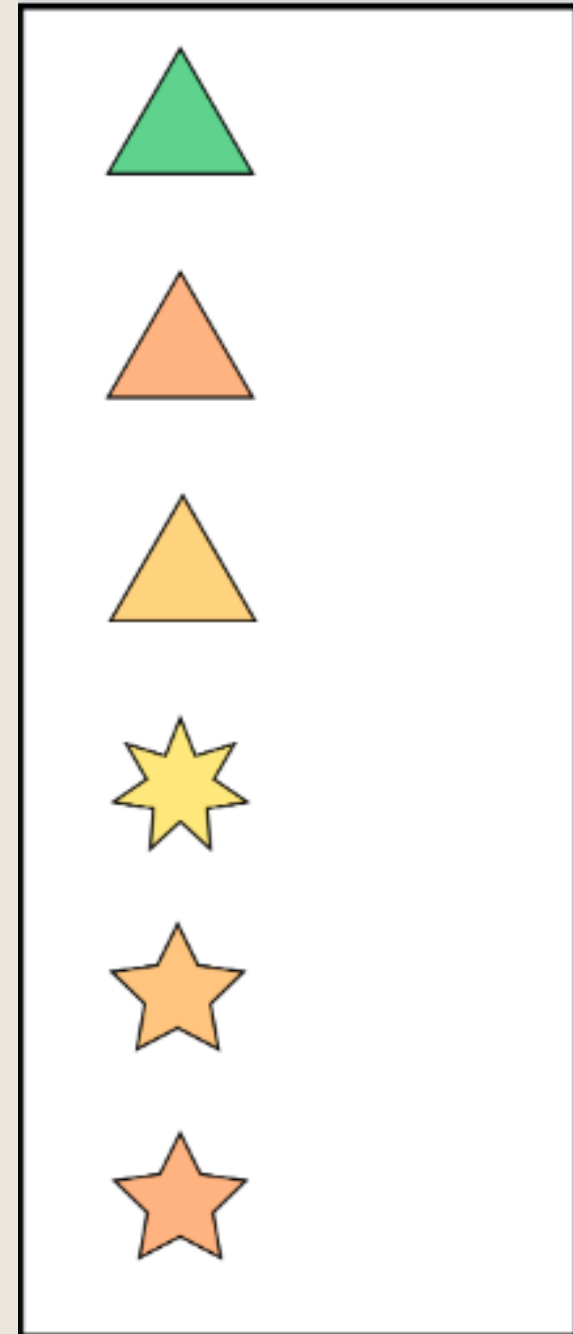
- Se agregan algunas variaciones en los hijos al azar.

NP.add(mutation(Hijo1))

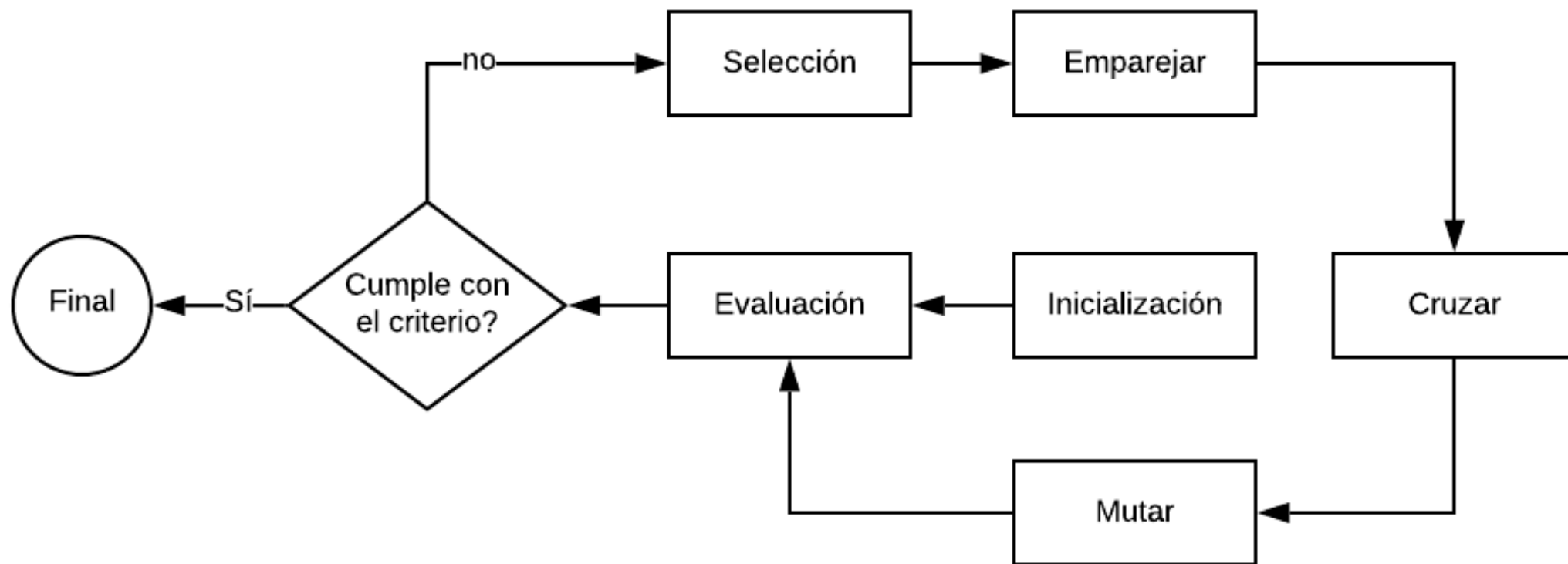
NP.add(mutation(Hijo2))

NP.add(mutation(Hijo3))

NP.add(mutation(Hijo4))



Evaluation



Tipos de selección

- **Selección proporcional:** Cuanto mas alto sea el fitness de un individuo mas alta será la probabilidad que pase al a siguiente generación.

$$m_s = \frac{m_o - \min(m_o)}{\max(m_o) - \min(m_o)}$$

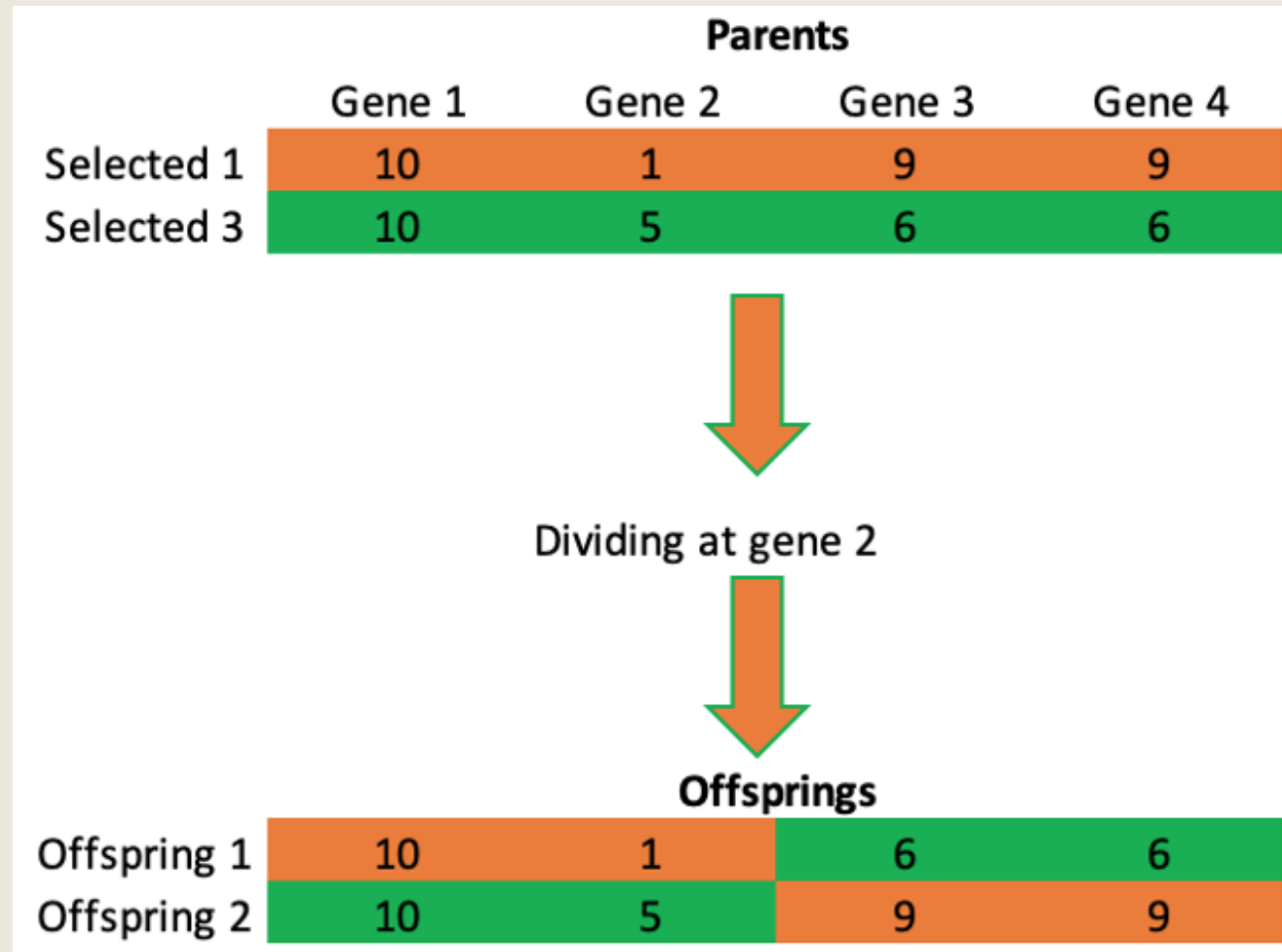
- **Selección por torneo:** Se eligen individuos al azar y el fitness mas alto del subgrupo es el que pasara a la siguiente generación.
- **Selección de la mejor mita:** La mitad de los mejores individuos pasan a la siguiente generación.
- **Selección al azar:** Los individuos son elegidos de forma aleatoria, con igual probabilidad para todos.

Tipos de emparejamiento

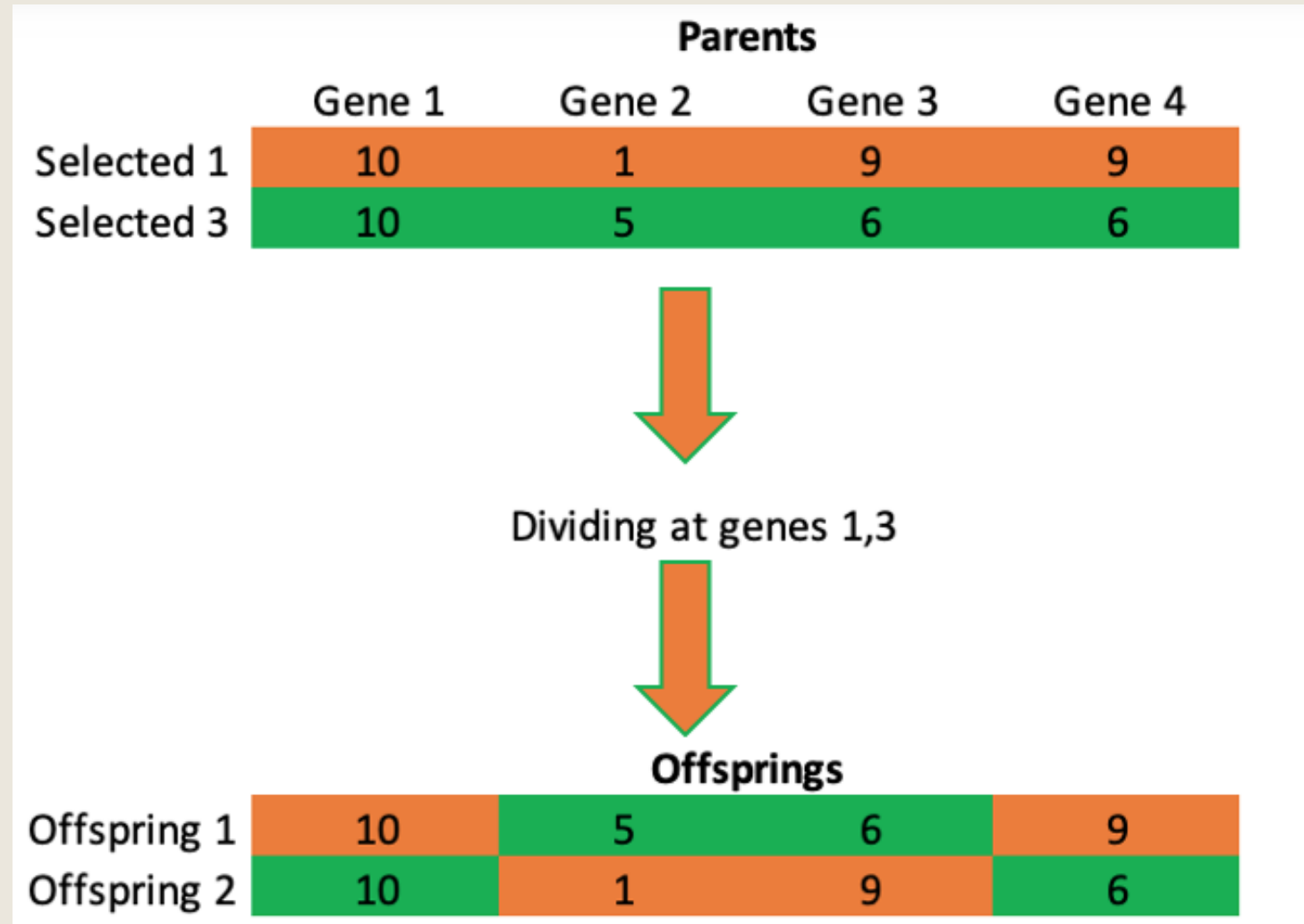
- **Mas apto:** Los mejores se emparejan con los mejores y los peores con los peores.
- **Aleatorio:** Las parejas se eligen de forma aleatoria.

Tipos de cruzamiento

- Unico Punto



- Dos puntos



- Uniforme

Cada gen tiene la probabilidad del 50% de ser de un padre u otro.

Parent 1	1	0	0	1	1
Parent 2	0	0	1	1	0
Offspring	1	0	1	1	0

Criterio de finalización

- Máximo fitness
- Máximo fitness promedio
- Numero máximo de generaciones
- Máximo números similares de fitness

Nota: cuidado al elegir su criterio de finalización, puede que el algoritmo nunca termine.

Algoritmo General

```
generacion = 0
poblacion = inicializarPoblacion()
fin = verificarCriterio(poblacion, generacion)

while(fin == None):
    padres = seleccionarPadres(poblacion)
    poblacion = emparejar(padres)
    generacion += 1
    fin = verificarCriterio(poblacion, generacion)
```