Escuela Superior de Cómputo
Instituto Politécnico Nacional

Algoritmos Genéticos

Dra. en C. Miriam Pescador Rojas

Antecedentes

- Los algoritmos genéticos (AG) fueron desarrollados originalmente por John Holland en 1975.

- Un AG es una **heurística de búsqueda que imita el proceso de evolución natural**

Utiliza conceptos de teorías de evolución tal como: **"Selección Natural"** y **"Herencia Genética".**

Introducción

- La principal metáfora de la computación evolutiva consiste en:
- La evolución de la naturaleza como fuente de inspiración (adaptación/selección natural)
- La aptitud de los individuos se determina por su medio ambiente

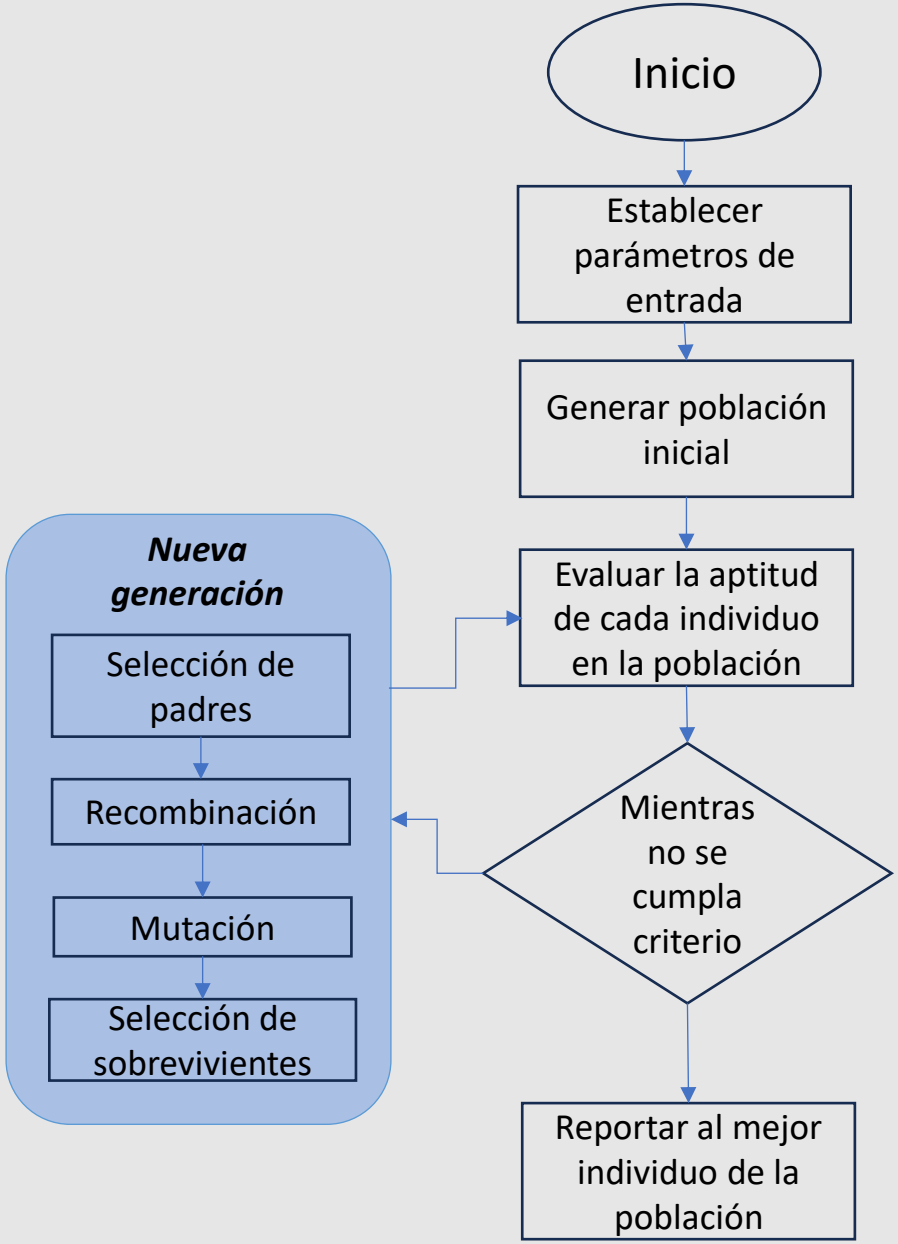
EVOLUCIÓN		**PROB. DE OPTIMIZACIÓN**
Ambiente	⟷	Problema
Individuo	⟷	Solución candidata
Población	⟷	Conjunto de soluciones
Aptitud	⟷	Calidad de la solución

Calidad → oportunidad de generar nuevas soluciones

Aptitud → posibilidades de supervivencia y reproducción

Esquema general de un algoritmo genético

- Dada una población de individuos, la presión ambiental causa una selección natural.
- Los individuos con mayor aptitud tienen una mayor probabilidad de reproducirse y heredar sus características genéticas.
- Se genera una nueva generación de individuos que heredaron características de sus padres.
- Solo sobreviven los individuos que mejor se adaptan al ambiente (elitismo).

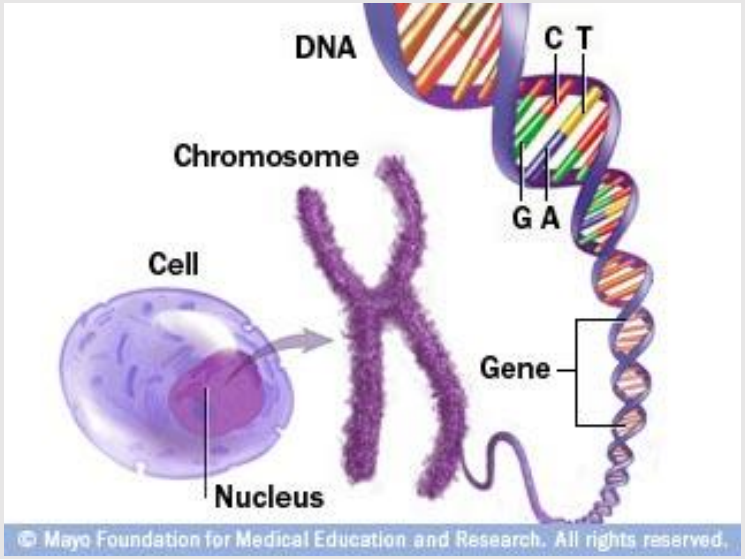


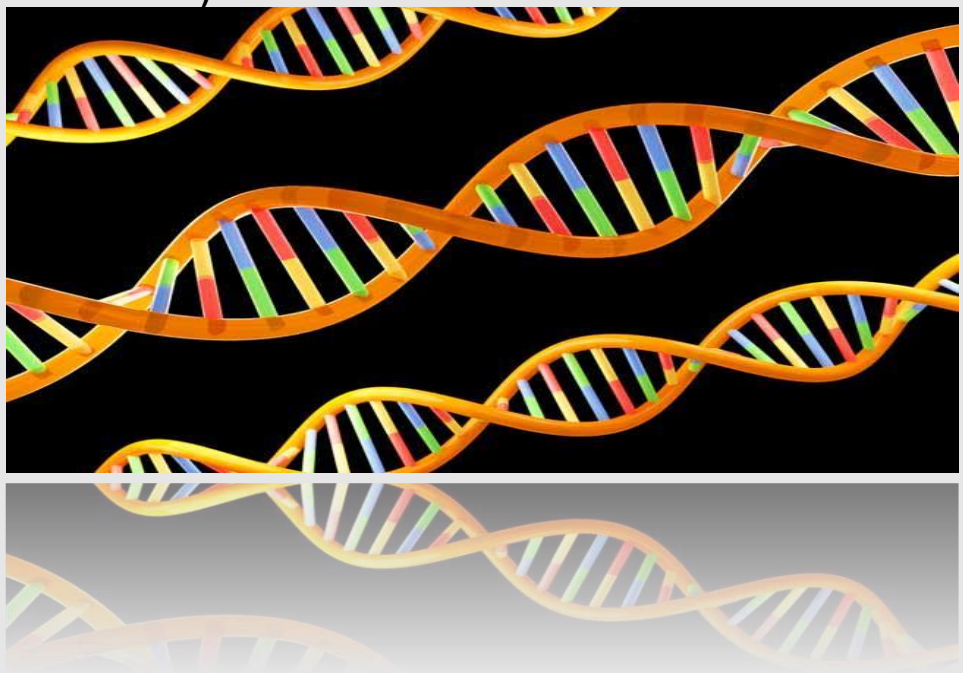
Algoritmo genético

- Crear una población inicial de individuos
- Evaluar la aptitud de los individuos
- Mientras **el criterio de terminación** no se cumpla, entonces:
  - Seleccionar a los padres para la recombinación
  - Recombinar (con cierta probabilidad) la información genética de los padres para generar descendientes
  - Aplicar (con cierta Probabilidad) proceso de mutación a los descendientes
  - Evaluar la aptitud de los nuevos individuos en la población
  - Seleccionar a los sobrevivientes para la siguiente generación (Elitismo)
- Reportar al mejor individuo de la población

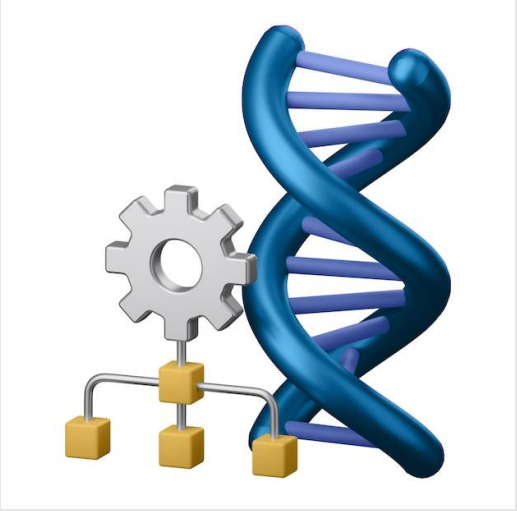
Codificación de la información genética

- La información necesaria para construir un organismo vivo está codificada en el ADN de ese organismo
- El genotipo (ADN en el interior) determina el fenotipo
- Genes → rasgos fenotípicos es un mapeo complejo
- Pequeños cambios en el genotipo conducen a pequeños cambios en el organismo (por ejemplo, altura, color del cabello)



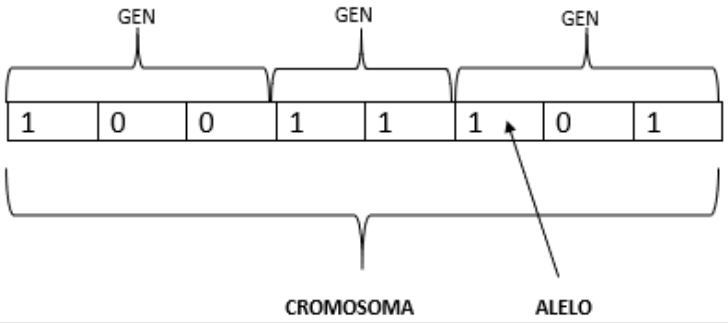


Tipos de codificación



- **Codificación binaria,**

Cromosoma A	101100101100101011100101
Cromosoma B	111111100000110000011111

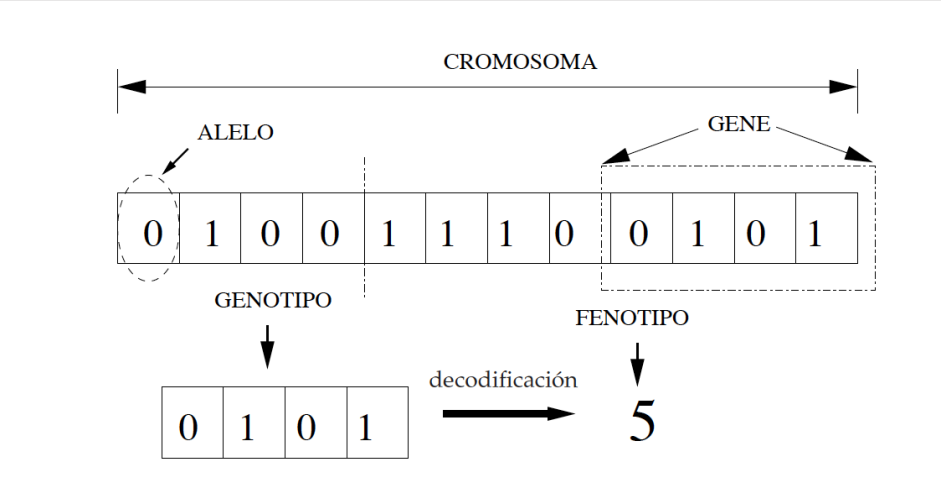


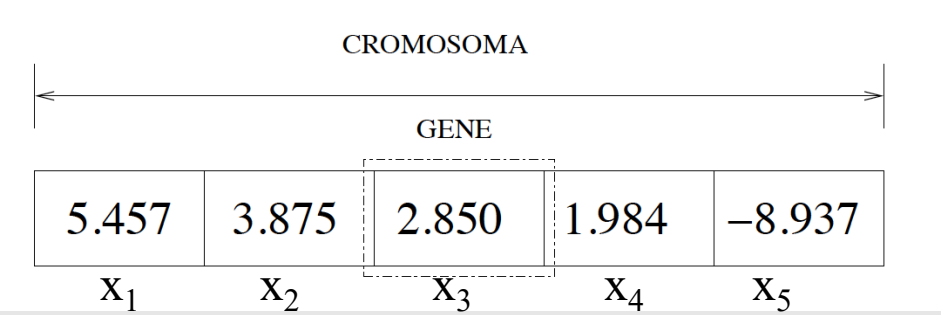
- **Codificación de permutaciones**

Chromosoma A	1 5 3 2 6 4 7 9 8
Chromosoma B	8 5 6 7 2 3 1 4 9

* Se considera un proceso de codificación y decodificación de las variables de un problema

Codificación del problema



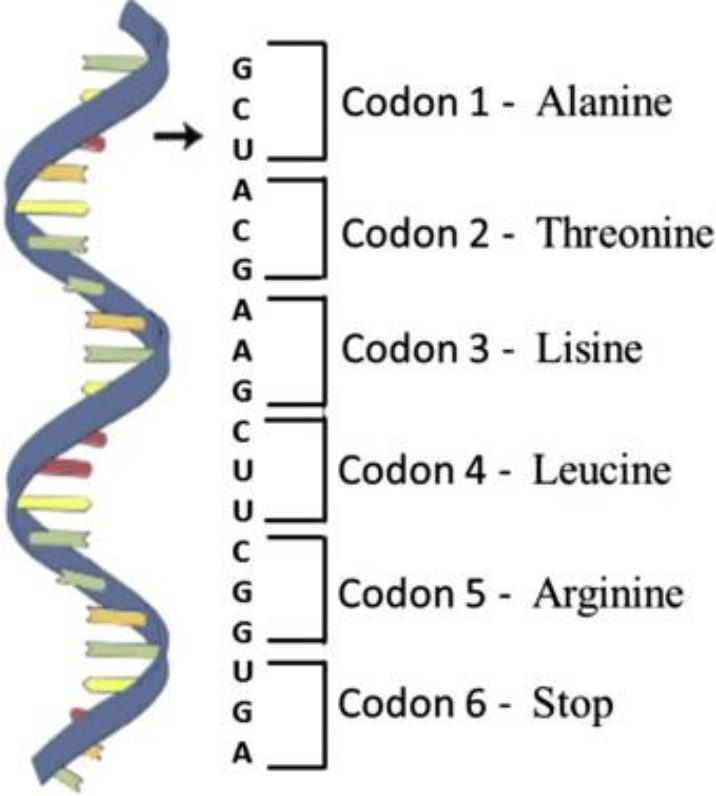


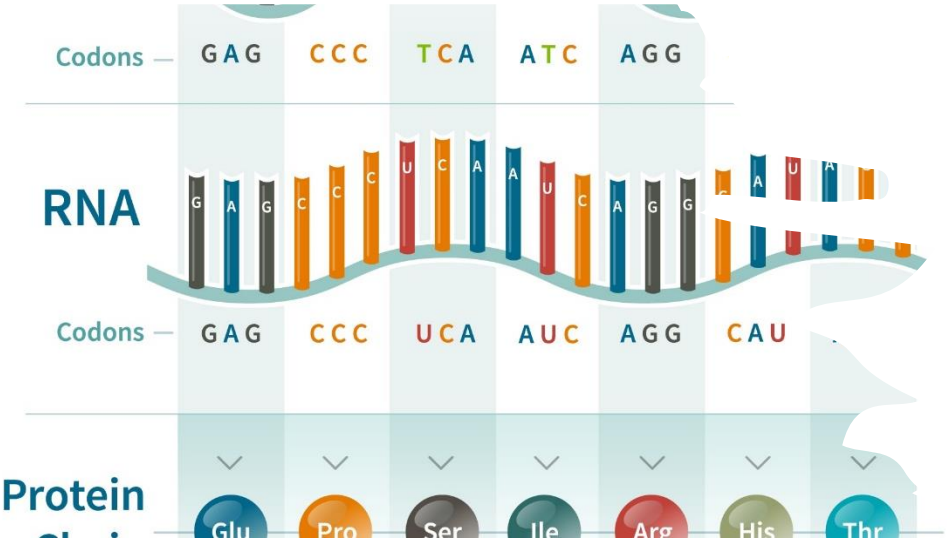
Tip. En cada generación, después de aplicar los operadores evolutivos verificar que las variables se encuentren dentro de los límites.

Tipos de codificación

- **Codificación directa de las variables de decisión**
  - **Representación real o entera**
  - **Simbólica**
  - **Por instrucciones.**

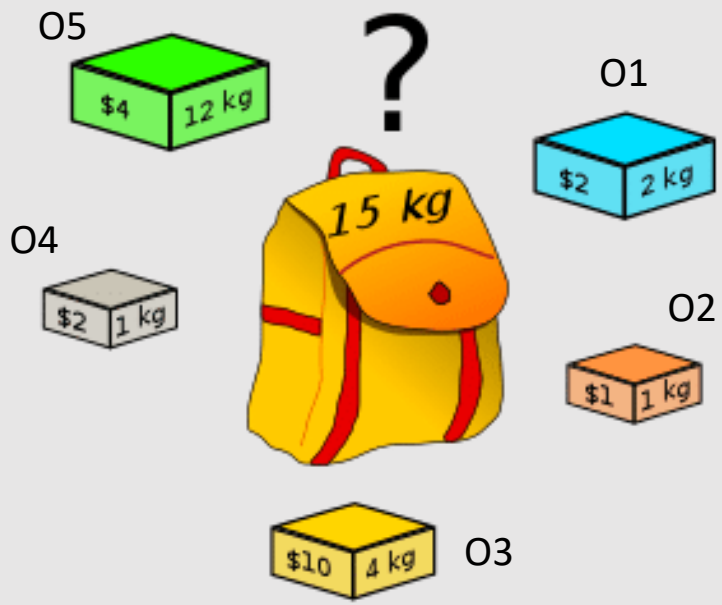
Cromosoma A	1.2324 5.3243 0.4556 2.3293 2.4545
Cromosoma B	ABDJEIFJDHDIERJFDLDFLFEGT
Cromosoma C	(atrás), (atrás), (derecha), (adelante), (izquierda)





Ejemplos de problemas con representación binaria

Optimización : El problema de la mochila (entera)



$$maximizar \quad \sum_{i=1}^{n} x_i \, v_i$$

$$sujeto\ a \quad \sum_{i=1}^{n} x_i \, w_i \leq W$$

$$donde: \quad v_i > 0, \; w_i > 0,$$

$$x_i \in \{0, 1\} \qquad para \quad 0 \leq i \leq n$$

Posibles soluciones candidatas

Id Solución	O1	O2	O3	O4	O5	f(x)	Fact?
1	1	0	1	0	1		
2	1	1	1	0	0		
3	1	0	0	1	1		
4	1	1	1	1	1		

Problema de minería de datos (modelado)

Reglas de asociación

T Id	Transacción
1	{pan, leche}
2	{pan, pañales, cerveza, huevo}
3	{leche, pañales, cerveza, cocacola}
4	{pan, leche, pañales, cerveza}
5	{pan, leche, pañales, cocacola}

El *soporte* de una regla:

$$supp(X \rightarrow Y) = supp(X \cup Y)$$

La *confianza* de una regla:

$$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$$

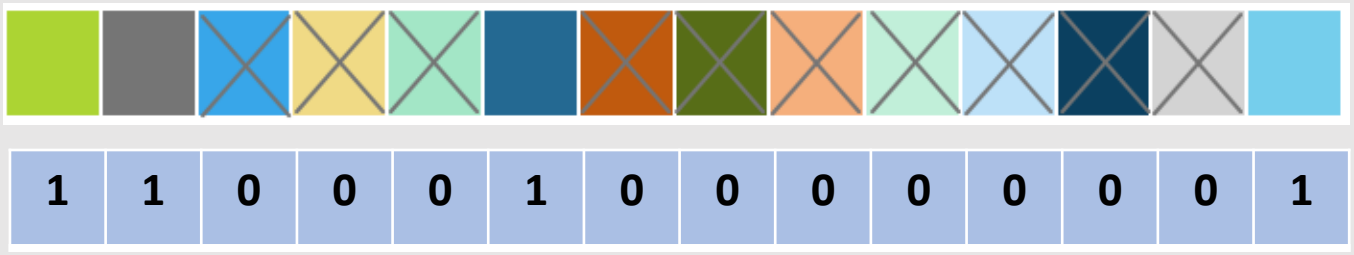
T Id	Cerveza	Pan	Leche	Pañales	Huevo	CocaCola
1	0	1	1	0	0	0
2	1	1	0	1	1	0
3	1	0	1	1	0	1
4	1	1	1	1	0	0
5	0	1	1	1	0	1

Problema de aprendizaje máquina (clasificación)

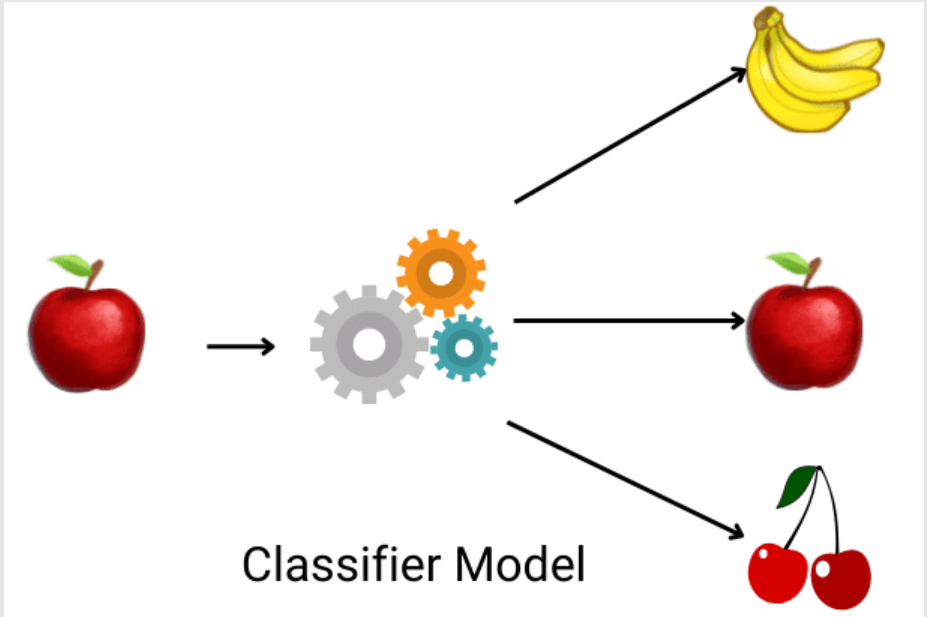
Selección de características

Todas las características

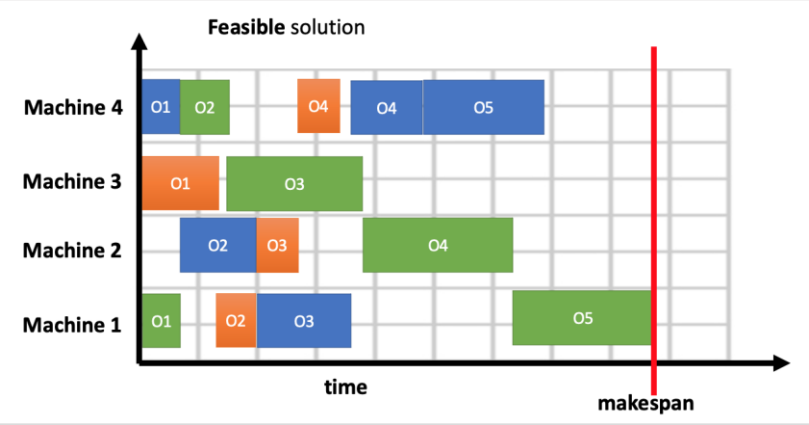
Selección de características

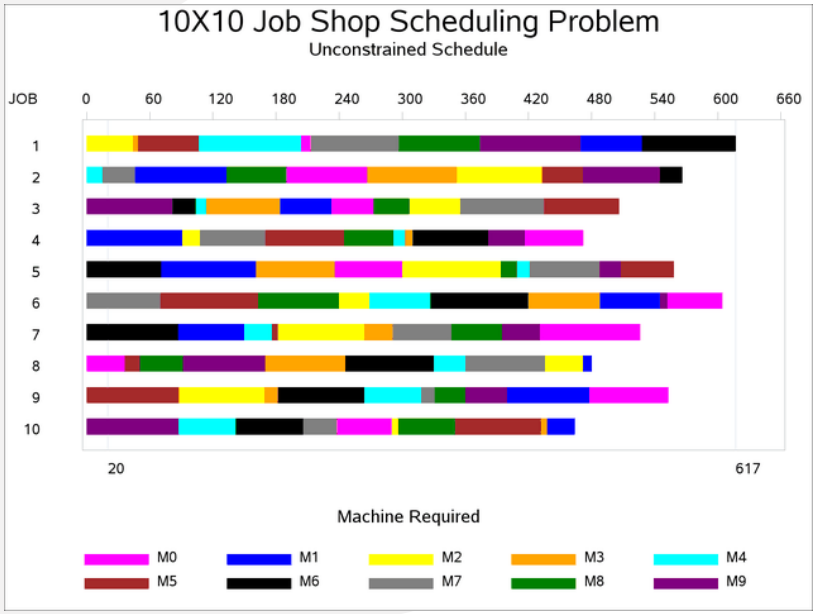


Características finales

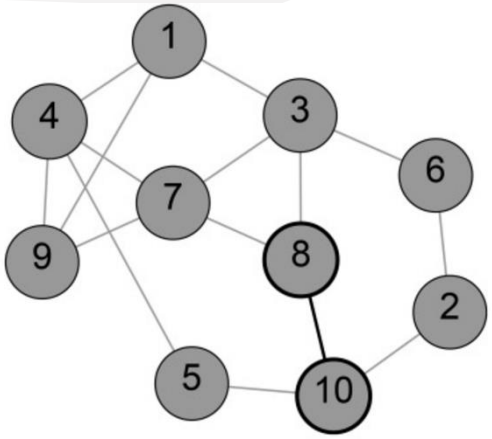


Problemas de planificación de tareas





0	0	0	0	1	0	
1	0	0	1	1	1	
1	1	0	0	1	0	
1	0	0	1	1	1	
1	0	1	1	0	1	
0	0	0	0	0	1	

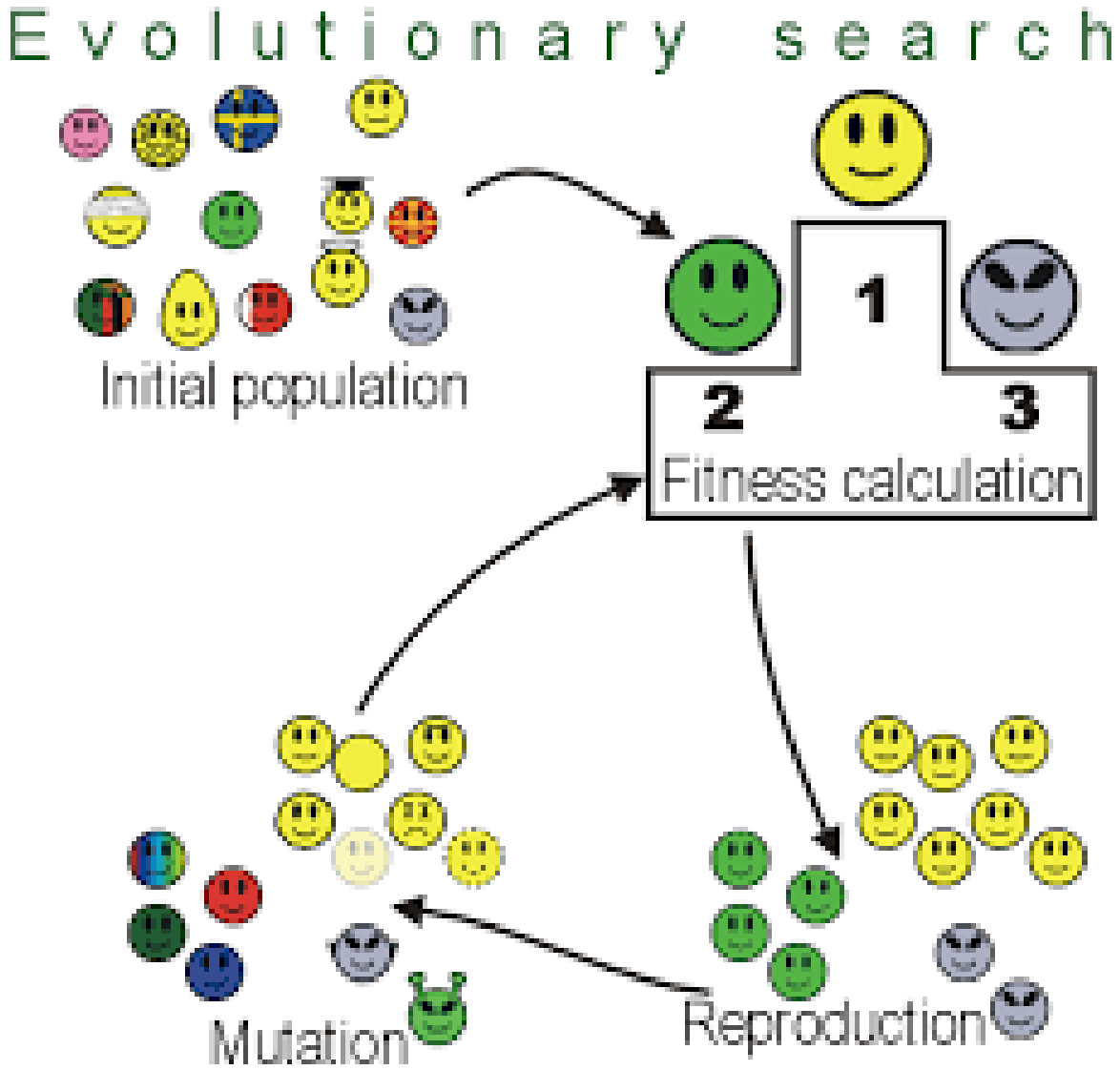


(a)

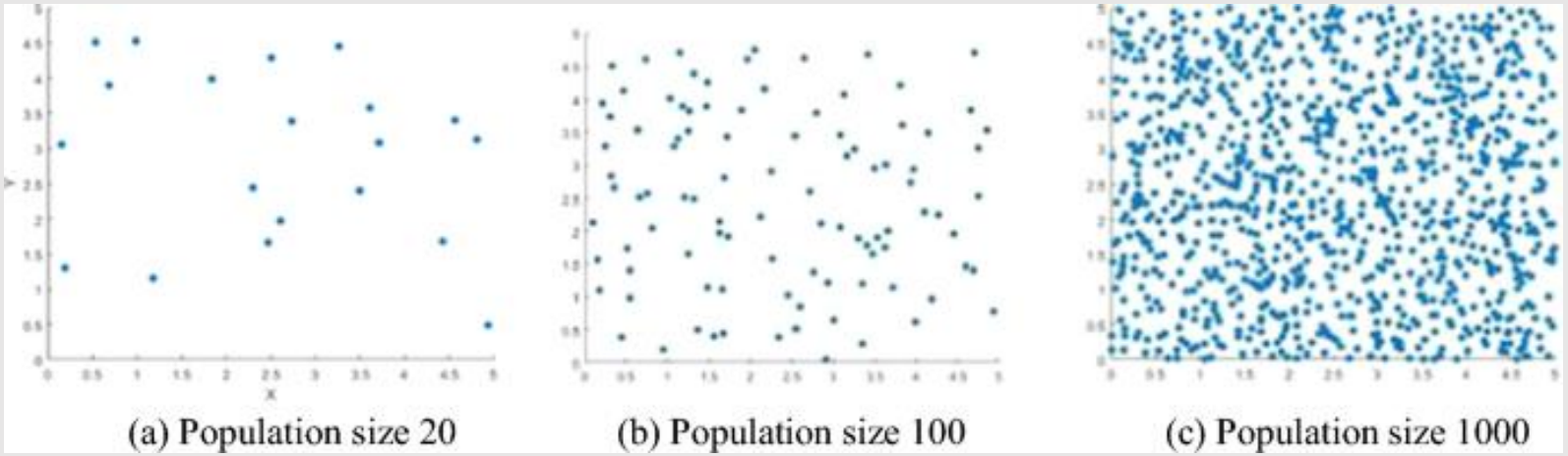
Nodes	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	0	0	0	0	1	0
2	0	0	0	0	0	1	0	0	0	1
3	1	0	0	0	0	1	1	1	0	0
4	1	0	0	0	1	0	1	0	1	0
5	0	0	0	1	0	0	0	0	0	1
6	0	1	1	0	0	0	0	0	0	0
7	0	0	1	1	0	0	0	1	1	0
8	0	0	1	0	0	0	1	0	0	(1)
9	1	0	0	1	0	0	1	0	0	0
10	0	1	0	0	1	0	0	(1)	0	0

(b)

Generación de la población inicial



POBLACION INICIAL



Cuando no se tiene información del espacio de búsqueda la población inicial se puede generar mediante las siguientes técnicas:

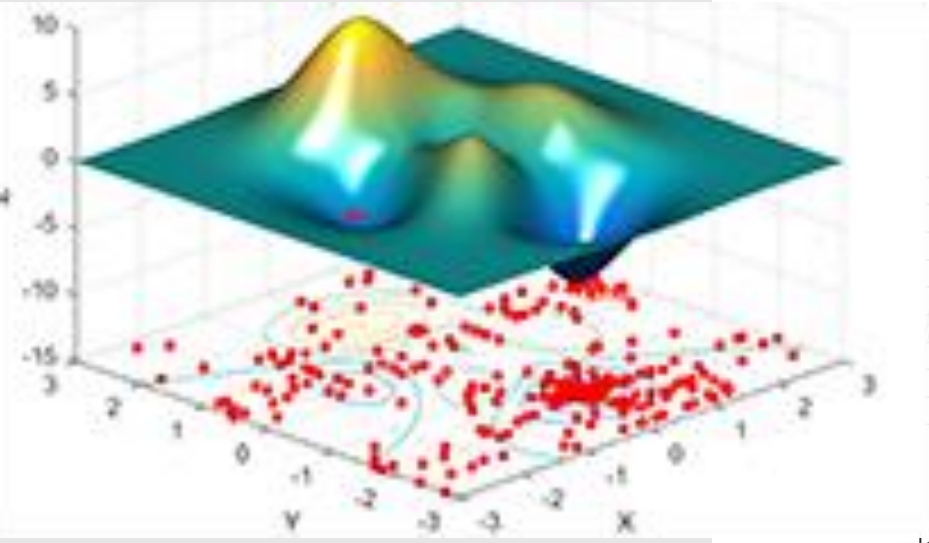
a) **Distribución uniforme aleatoria.**

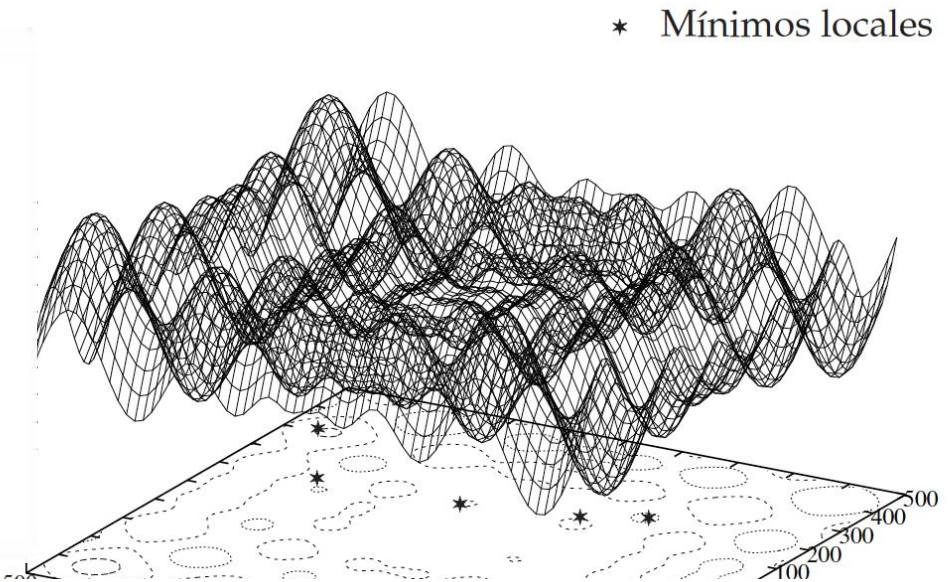
b) Diseño de experimentos, p.ej. **Hipercubos latinos**

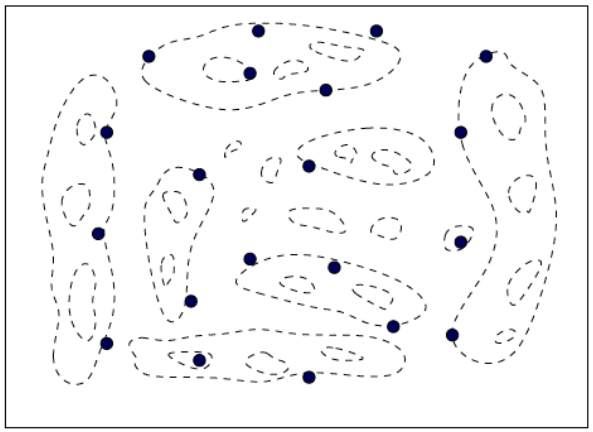
Por el contrario si se tiene conocimiento de **soluciones prometedoras** en el espacio de búsqueda es recommendable incorporarlas en la población inicial.

Población aleatoria en un problema de optimización continua

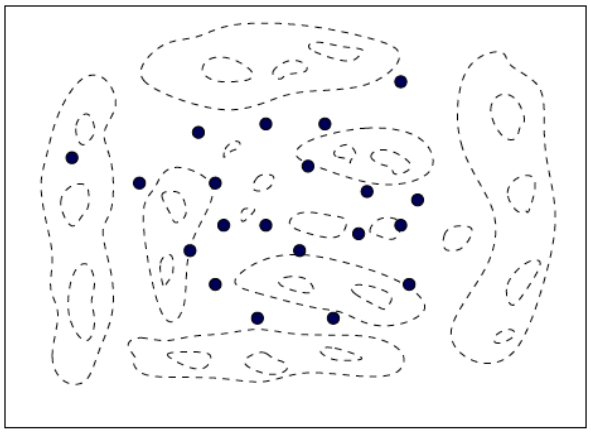
Paisaje de aptitud



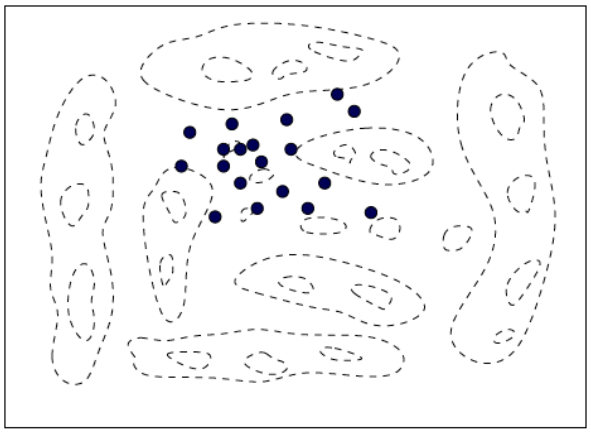




generación 1



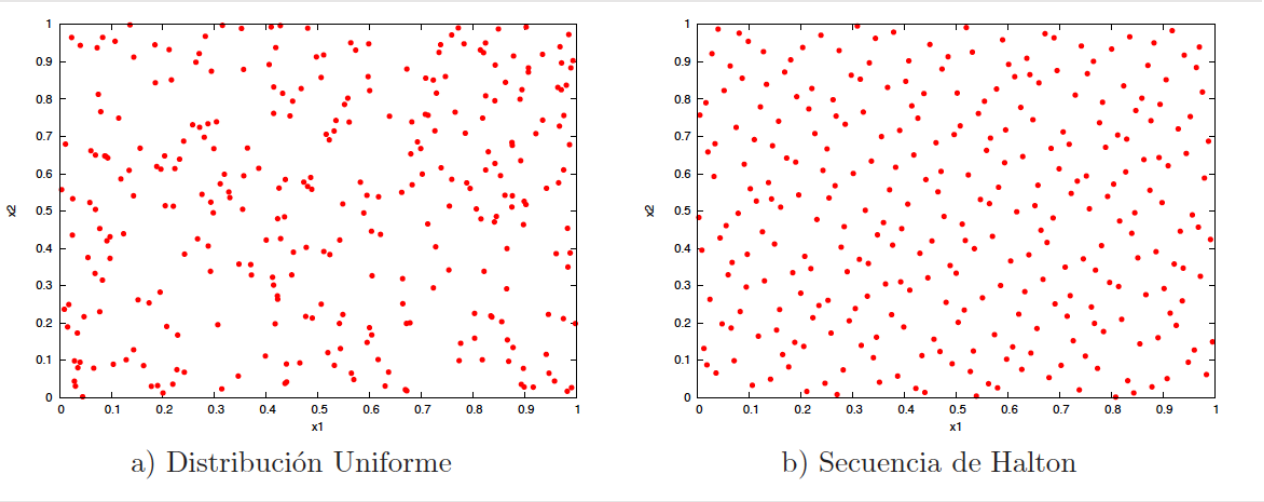
generación 50



generación 100

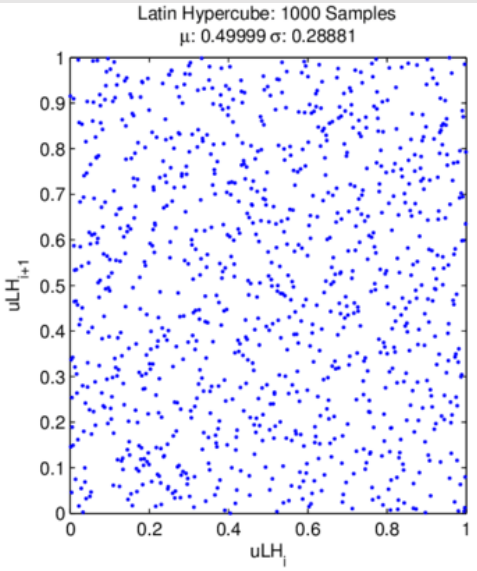
Figura 2.9: Ilustración gráfica del proceso evolutivo de un algoritmo genético.

Población inicial: Ejemplos de distribuciones



a) Distribución Uniforme

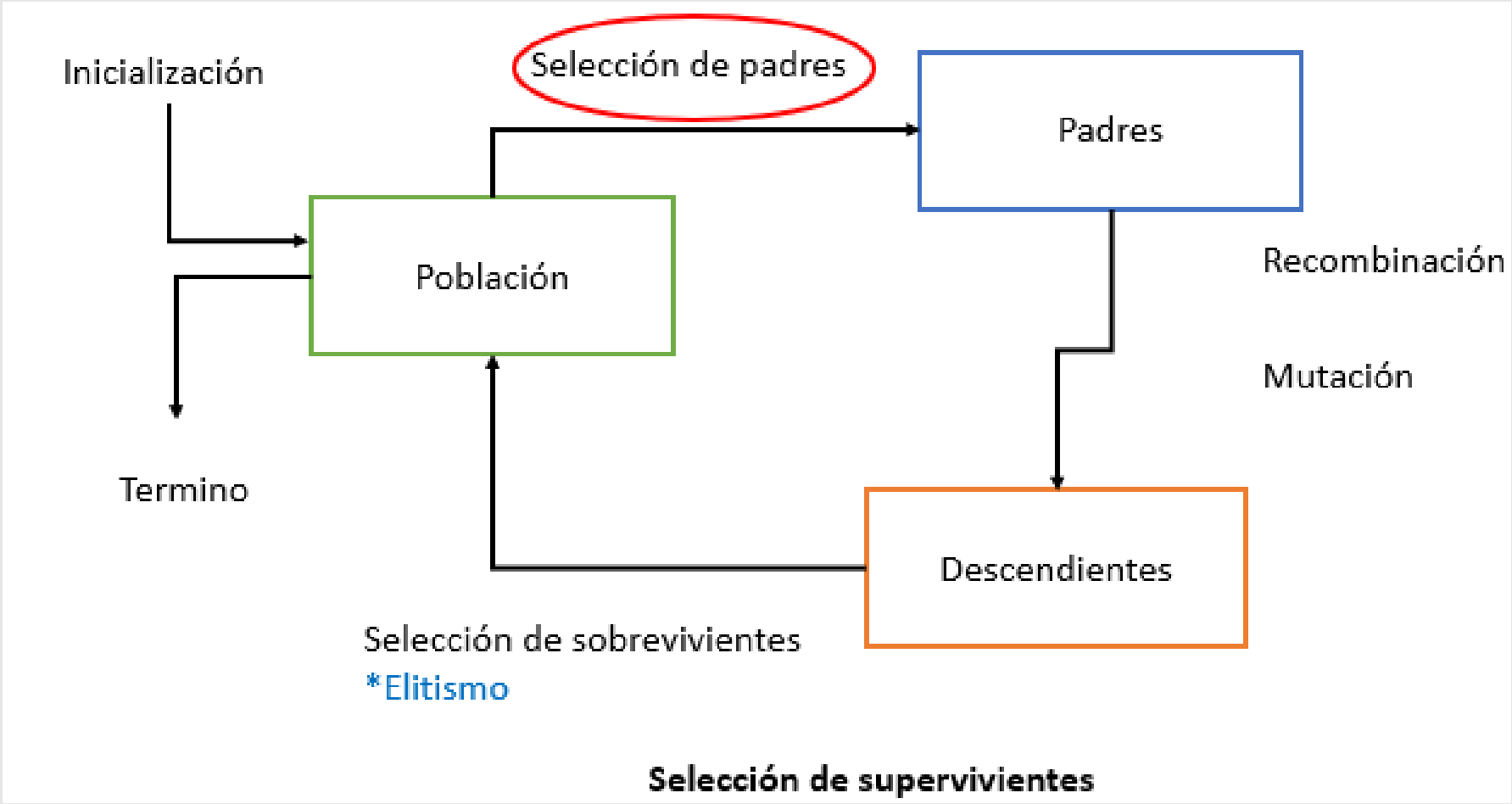
b) Secuencia de Halton



Hipercubos latinos

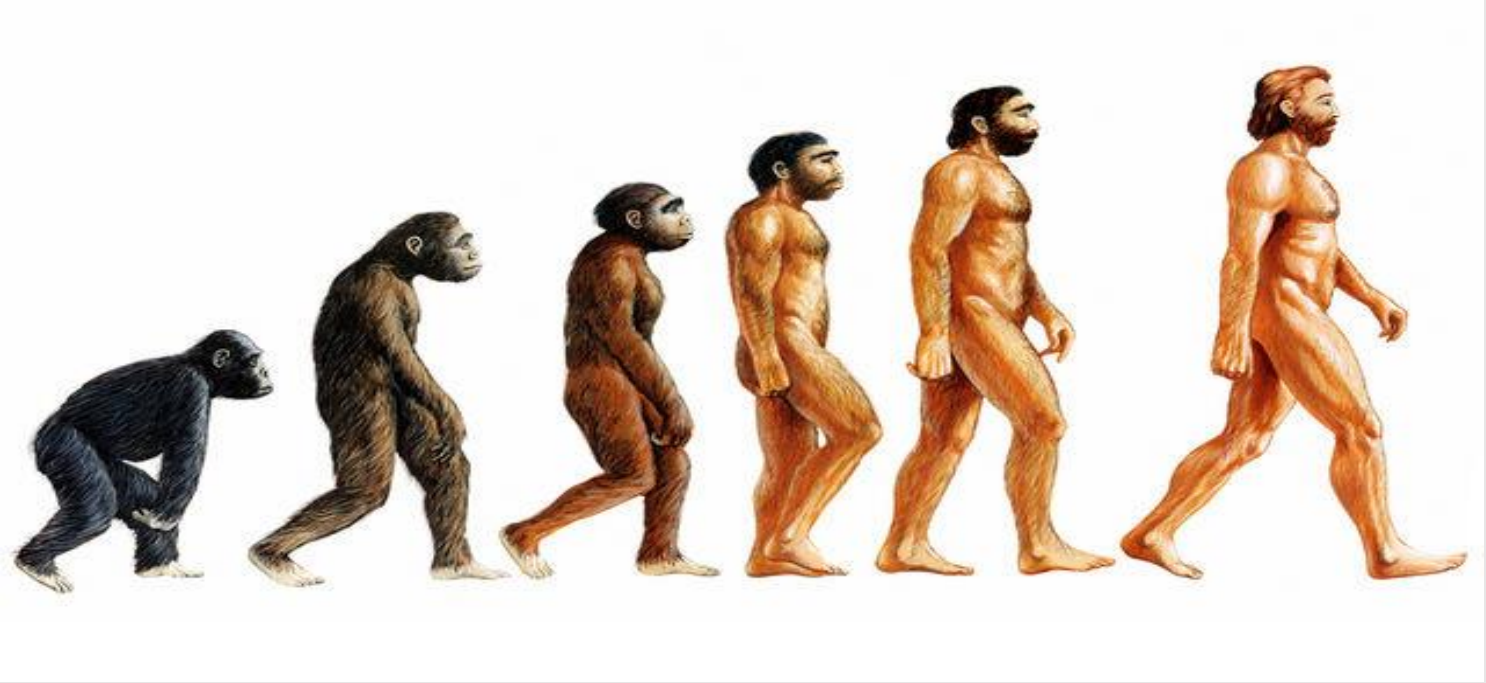
Tip. Introducir conocimiento del espacio de búsqueda.
Aproximaciones a la solución (por otros métodos)

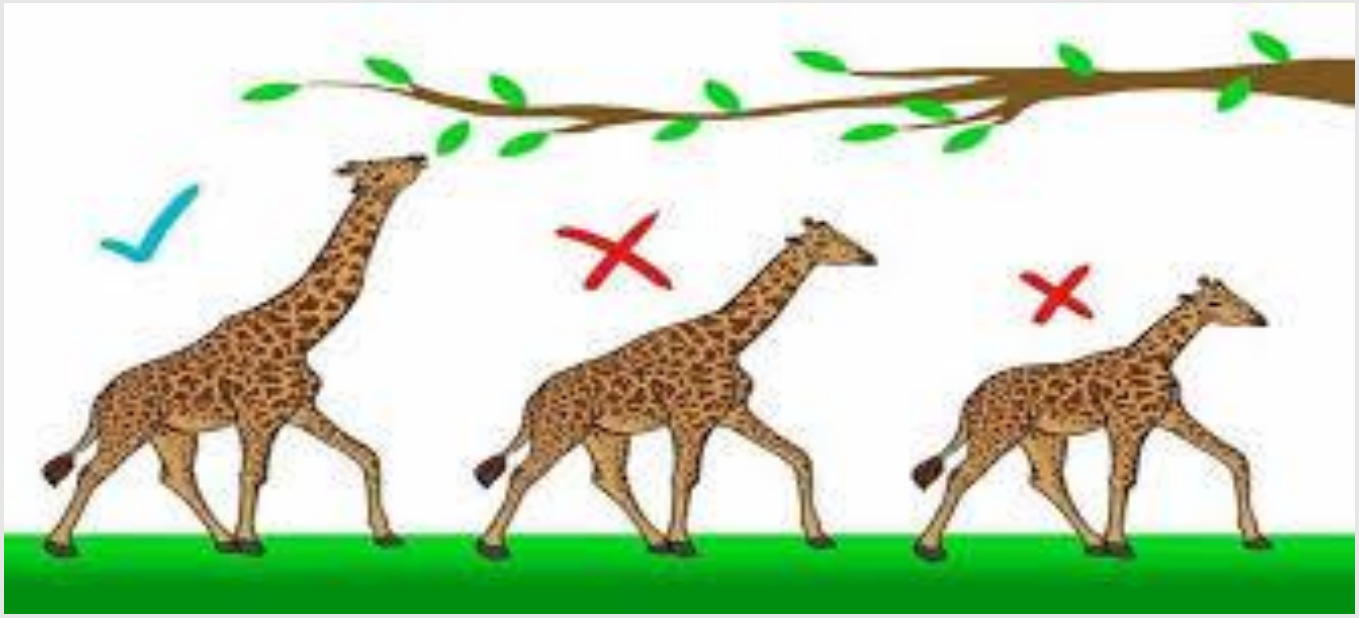
Esquema General de Algoritmo Genético

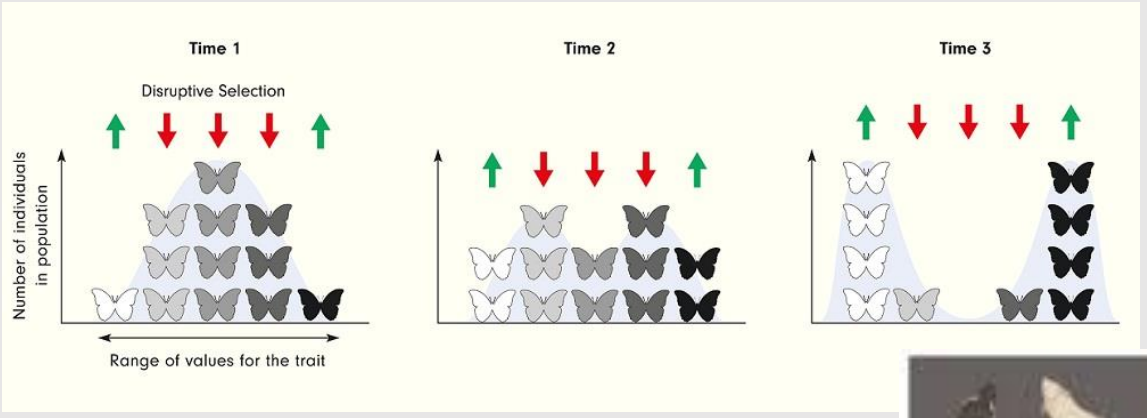


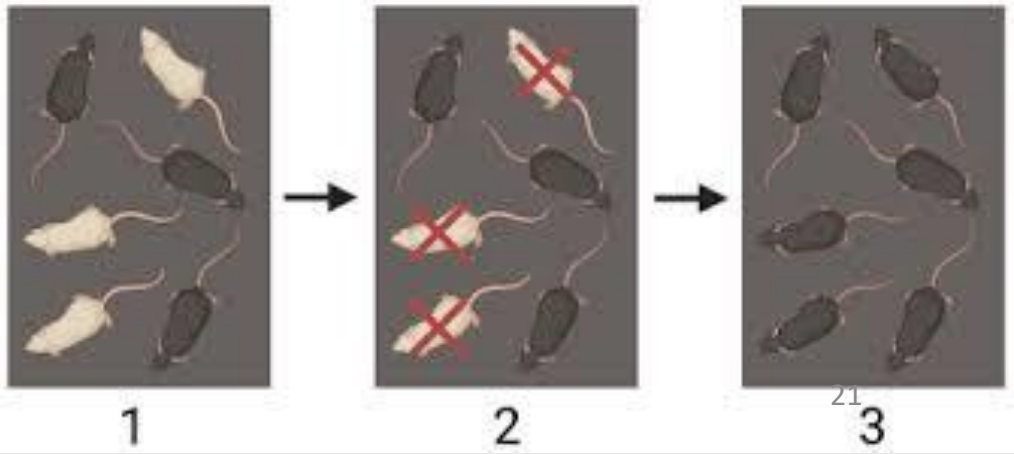
Selección Natural

En la teoría de la evolución de Darwin se dice que sólo los organismos **mejor adaptados** a su entorno tienden a **sobrevivir** y **transmitir sus características genéticas** en número creciente a las generaciones siguientes, mientras que los menos adaptados tienden a ser eliminados.






Time 1
Time 2
Time 3
Disruptive Selection
Number of individuals in population
Range of values for the trait


1
2
3

Selección de padres

- Muchos esquemas son posibles siempre y cuando los cromosomas de mejor puntuación sean más probables.

La puntuación se denomina a menudo como “aptitud” (fitness)

- Se puede utilizar diferentes estrategias de selección, entre las más comunes se tiene la selección por torneo.

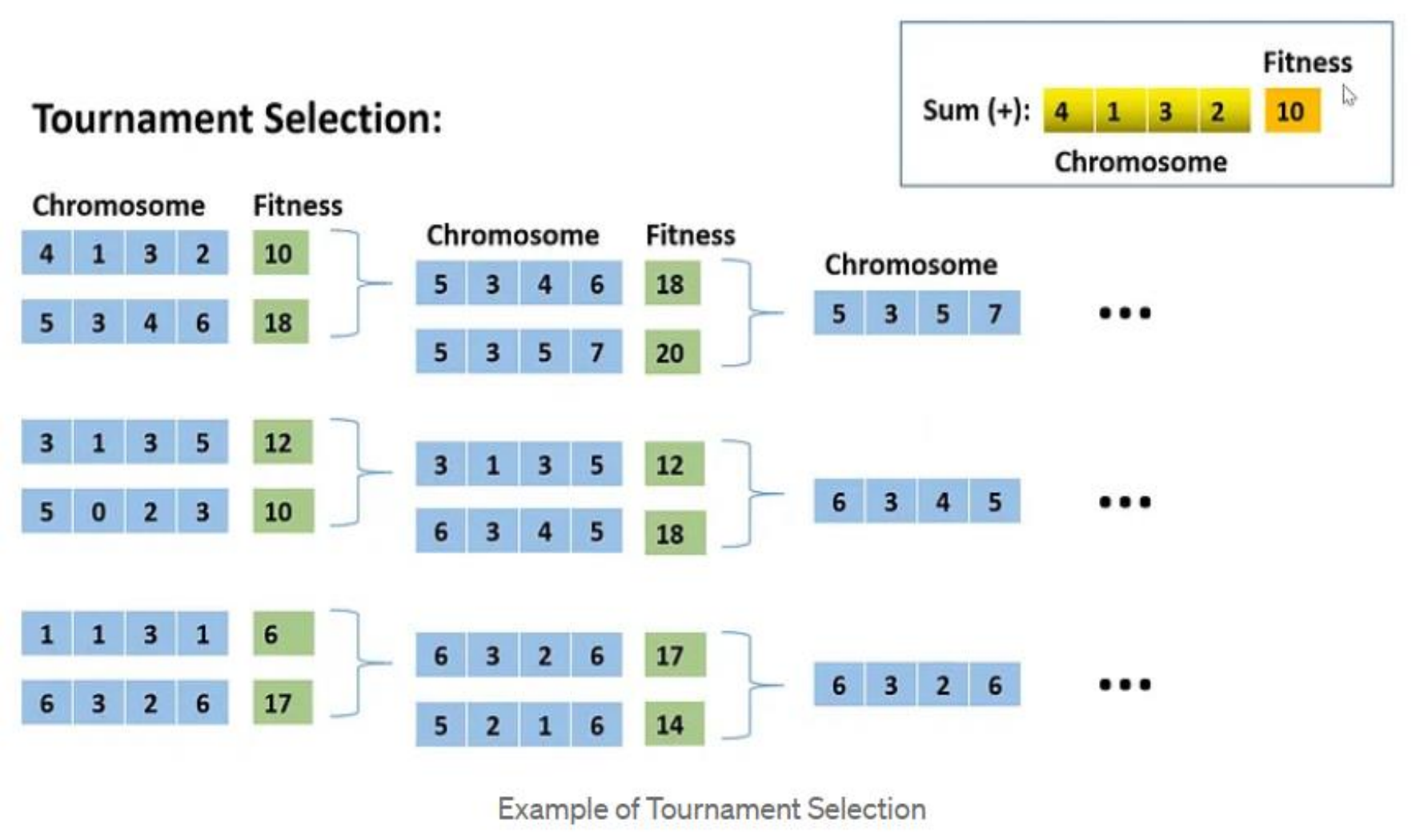
Selección por torneos

- Es un método para seleccionar individuos de una población en un algoritmo genético.
- Este método de selección implica la ejecución de varios "torneos" entre unos pocos individuos elegidos al azar de la población.
- El ganador de cada torneo (el que tiene la mejor aptitud) se selecciona para el proceso de recombinación.
- La presión de selección se ajusta fácilmente cambiando el tamaño del torneo. **Si el tamaño del torneo es mayor, los individuos más débiles tienen una probabilidad menor de ser seleccionados.**

Algoritmo para selección por torneo binario

- Mientras no se tenga el número total de individuos:
  - Se mezcla a los individuos de la población
  - Se eligen pares de individuos que competirán entre sí
  - Se selecciona al individuo con una mayor aptitud en cada torneo
- Veáse el siguiente ejemplo

Ejemplo de Torneo binario



Example of Tournament Selection

Torneo binario

Población actual

No	Aptitud
(1)	254
(2)	47
(3)	457
(4)	194
(5)	85
(6)	310

Torneo 1

Mezcla 1	Ganadores
(2)	
(6)	(6)
(1)	
(3)	(3)
(5)	
(4)	(4)

Torneo 2

Mezcla 2	Ganadores
(4)	
(1)	(1)
(6)	(6)
(5)	
(2)	
(3)	(3)

Individuos seleccionados y parejas conformadas:

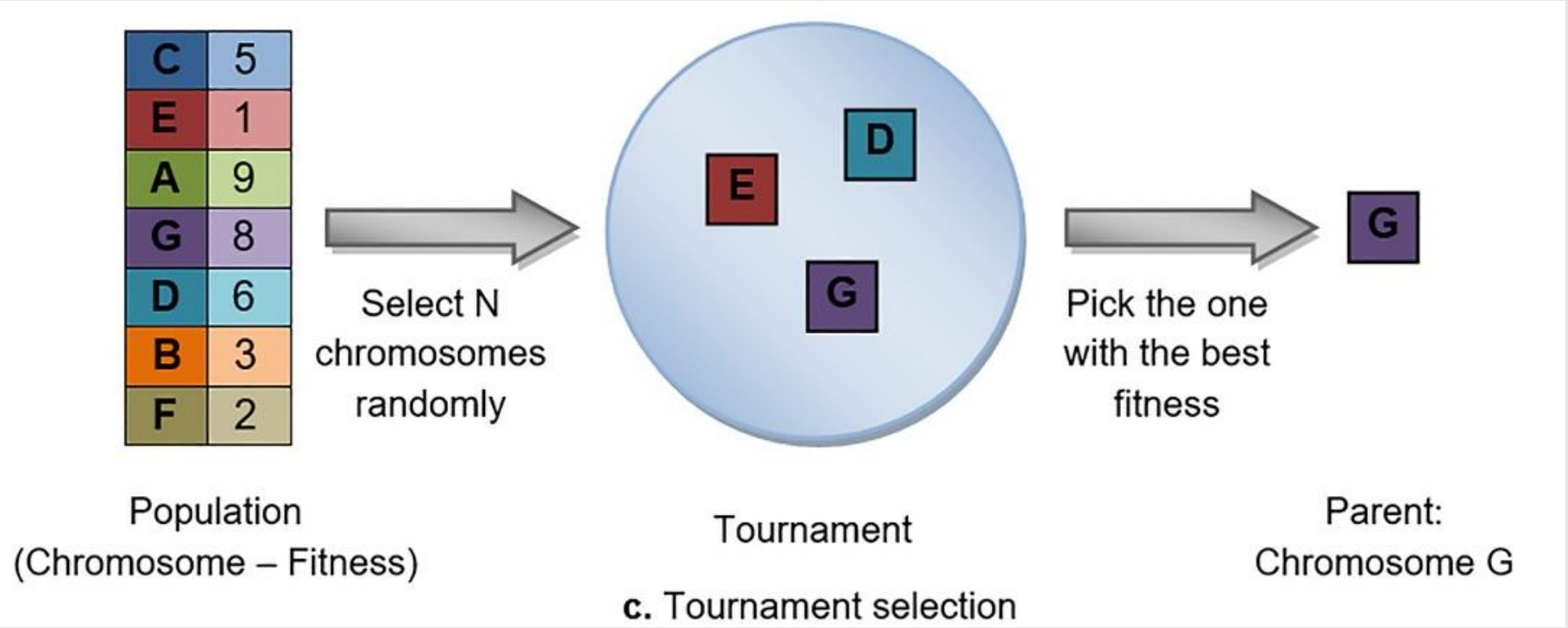
Pareja 1: (6) y (1), Pareja 2: (3) y (6), Pareja 3: (4) y (3)

Ejercicio

No	Fitness
(1)	111
(2)	695
(3)	354
(4)	998
(5)	210
(6)	15
(7)	96
(8)	152
(9)	348
(10)	563

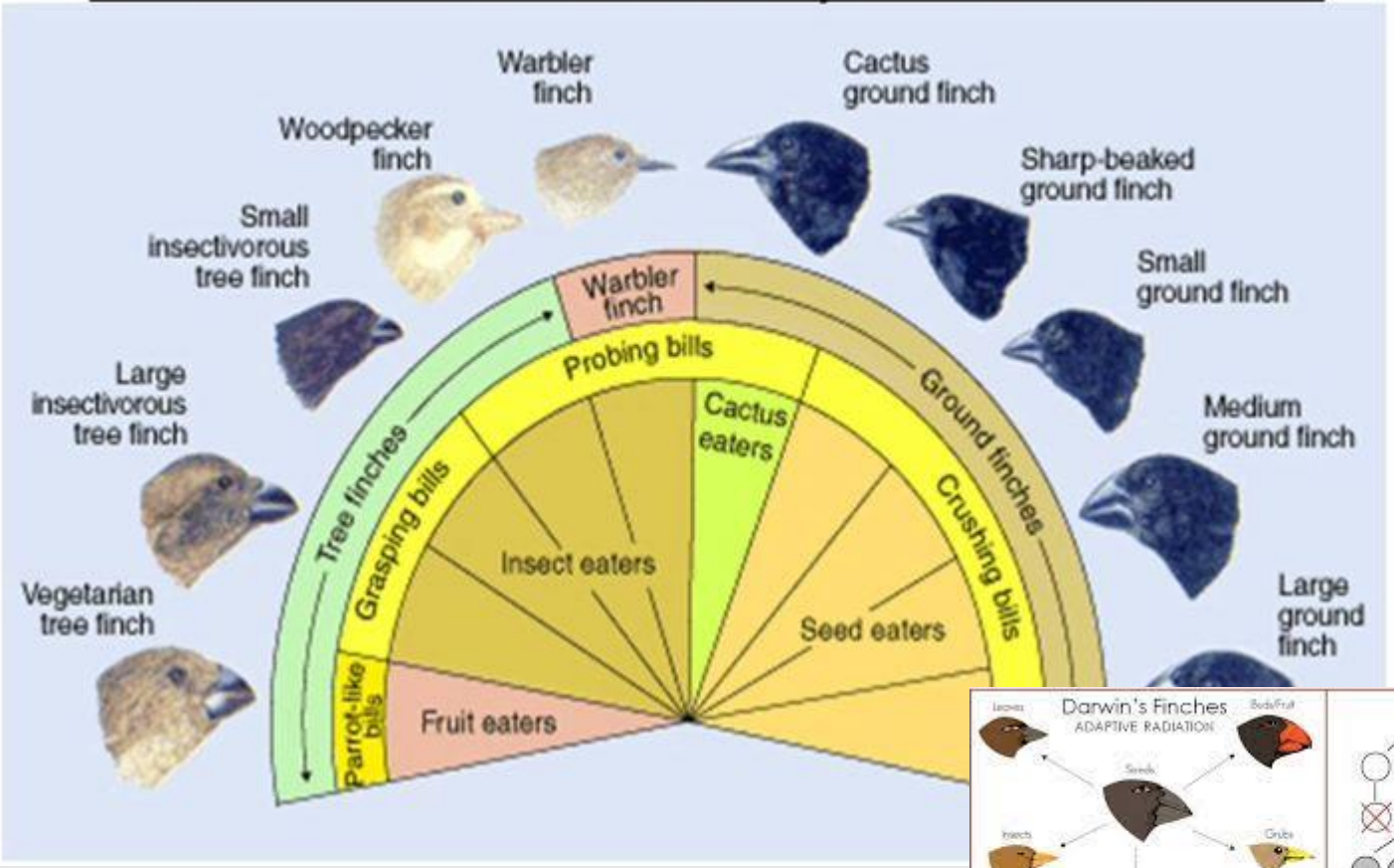
Torneo 1		Torneo 2	
Mezcla 1	Ganadores	Mezcla 2	Ganadores
(10)		(7)	
(1)		(5)	
(7)		(10)	
(2)		(1)	
(5)		(3)	
(8)		(8)	
(6)		(2)	
(3)		(4)	
(9)		(6)	
(4)		(9)	

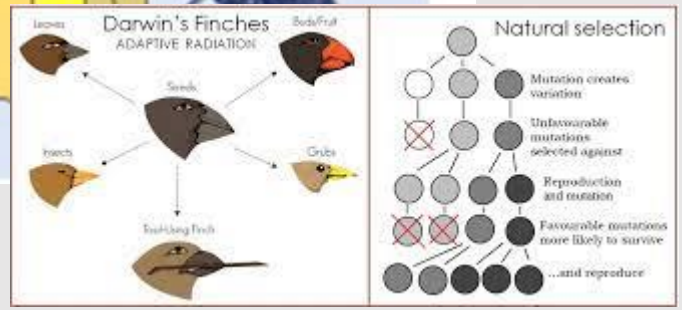
Ejemplo de Torneo n-ario



c. Tournament selection

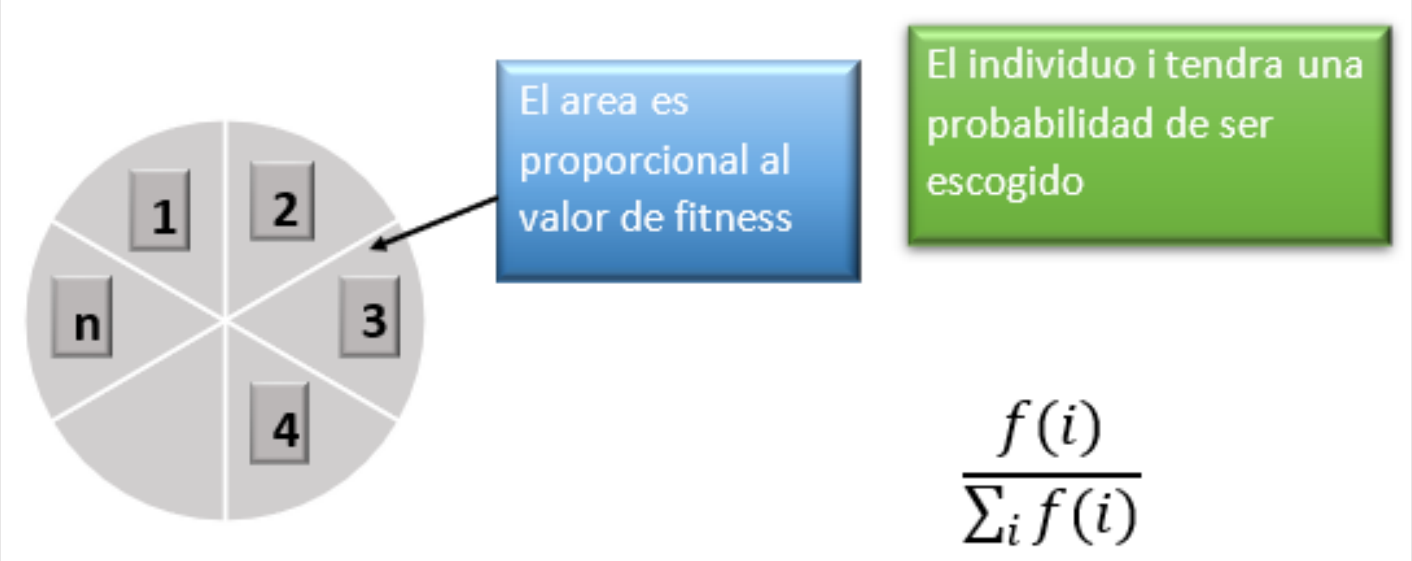
Allopatric and sympatric speciation of finches by natural selection into many different niches





Selección proporcional

Aplicamos la selección proporcional de aptitud con el método de ruleta:

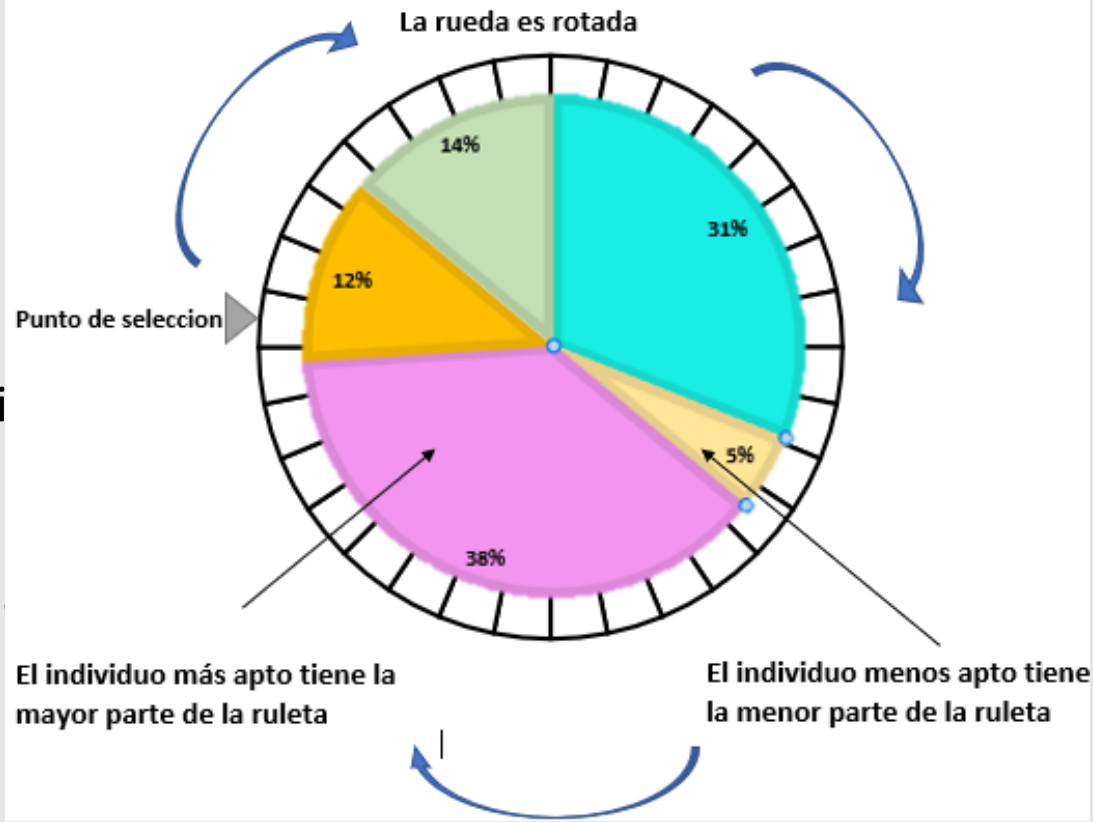


- Repetimos la extracción tantas veces como sea necesario (el tamaño de la población)

Selección de ruleta

Calcular el valor esperado T
• Repetir N veces (donde n es el tamaño de la población):

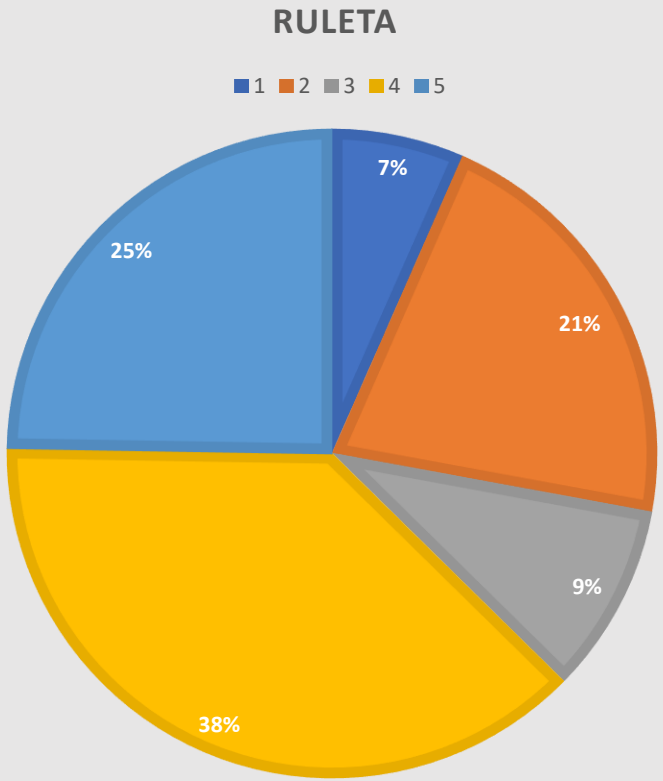
- Generar un número aleatori r entre 0.0 y T
- Agregue los valores esperados hasta que la sum sea mayor que igual a r
- Se selecciona al sujeto que hace esta suma exceda el límite.



Selección de ruleta

Ejemplo

id	Aptitud	Esperanza	Porcentaje
1	25	0.32894	6.57894
2	81	1.06578	21.31578
3	36	0.47368	9.473684
4	144	1.89473	37.89473
5	94	1.23684	24.73684
suma	**380**	**5.0**	**100**



Promedio 380 / 5 = 76
Generar r ∈ [0.0, 5.0]
r = 1.3
(ind1) sum = 0.33 < r
(ind2) sum = 1.39 > r
Individuo 2 es seleccionado

Ejercicio

No	Fitness	Valor esperado
(1)	111	
(2)	695	
(3)	354	
(4)	998	
(5)	210	
(6)	15	
(7)	96	
(8)	152	
(9)	348	
(10)	563	

Número aleatorio r ∈ [0.0, 10.0]
r = 5.5

Operadores de cruza (crossover)

- Para cada pareja, se decide con una cierta probabilidad (p. ej, 0.6) sí se recombinaran para generar nuevos descendientes.

- P. ej. suponga que solo se decidió recombinar las parejas (s_1, s_2) and (s_5, s_6).

- Para cada pareja, se aplica un operador de recombinación. Por ejemplo: cruza de un punto

Cruza de un punto en codificación binaria (pareja 1)

Padres antes de la cruza:

S_1 =

1	1	1	1	**0**	**1**	**0**	**1**	**0**	**1**

S_2 =

1	1	0	0	**1**	**1**	**0**	**1**	**0**	**1**

Descendientes después de la cruza

S1’ =

1	1	1	1	1	1	0	1	0	1

S2’’ =

1	1	0	0	0	1	0	1	0	1

Cruza de un punto en codificación binaria (pareja 2)

Antes de la cruza:

S_5 =

1	0	0	1	1	**1**	**1**	**0**	**0**	**0**

S_6 =

0	1	0	0	1	**1**	**1**	**1**	**0**	**0**

Después de la cruza

S5’ =

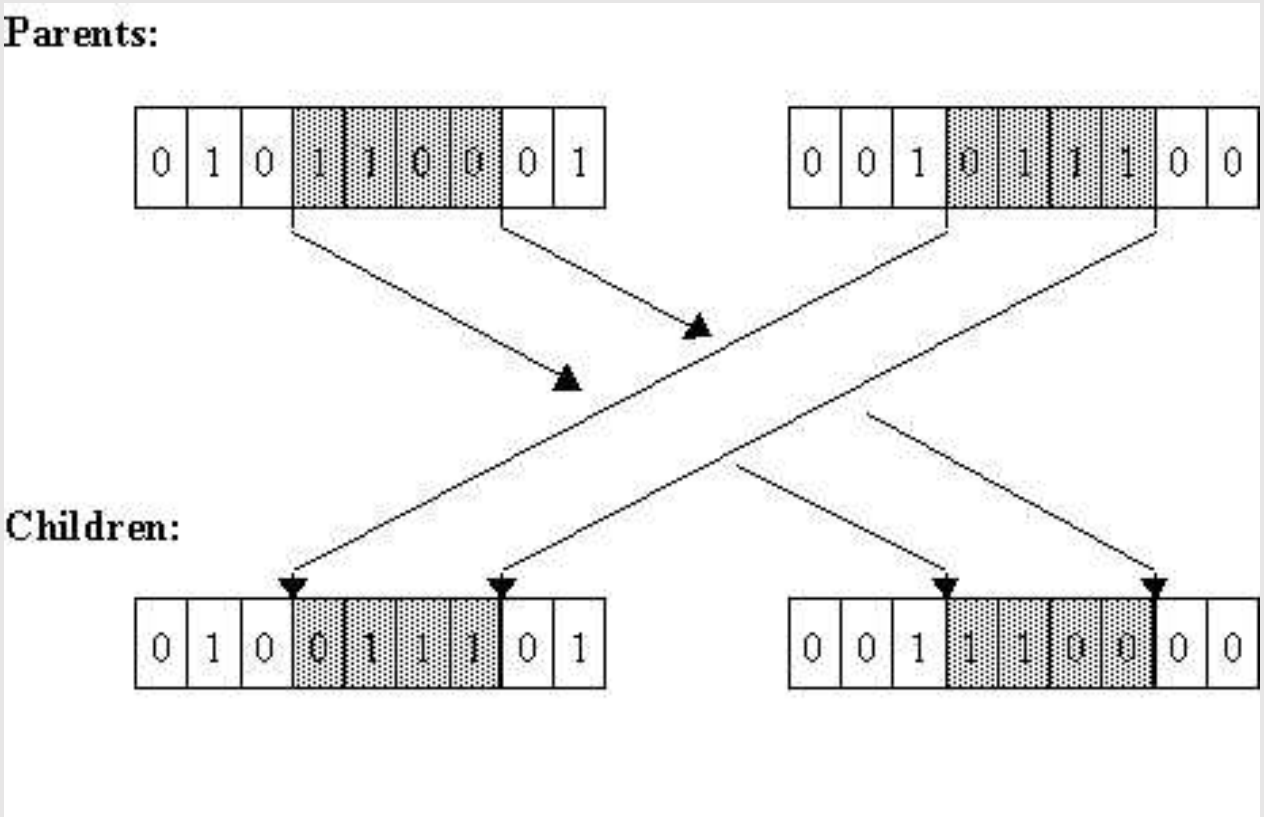
0	1	0	0	1	**1**	**1**	**0**	**0**	**0**

S6’ =

1	0	0	1	1	**1**	**1**	**1**	**0**	**0**

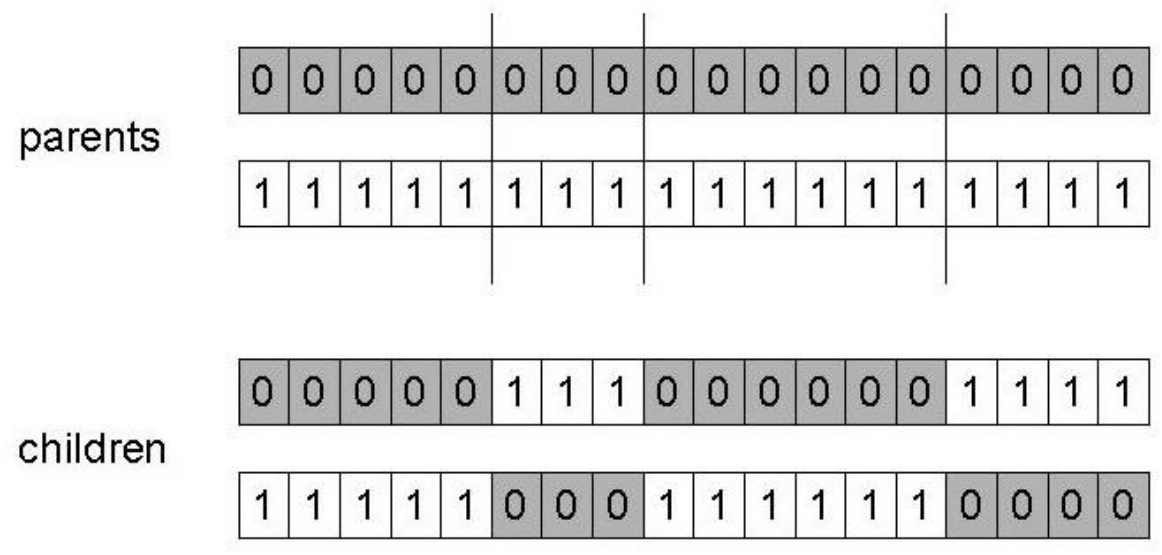
Cruza de 2 puntos

- Se seleccionan aleatoriamente 2 puntos de cruza
- Se realiza la división de la información con esos puntos
- Se intercambia la información genética como sigue:



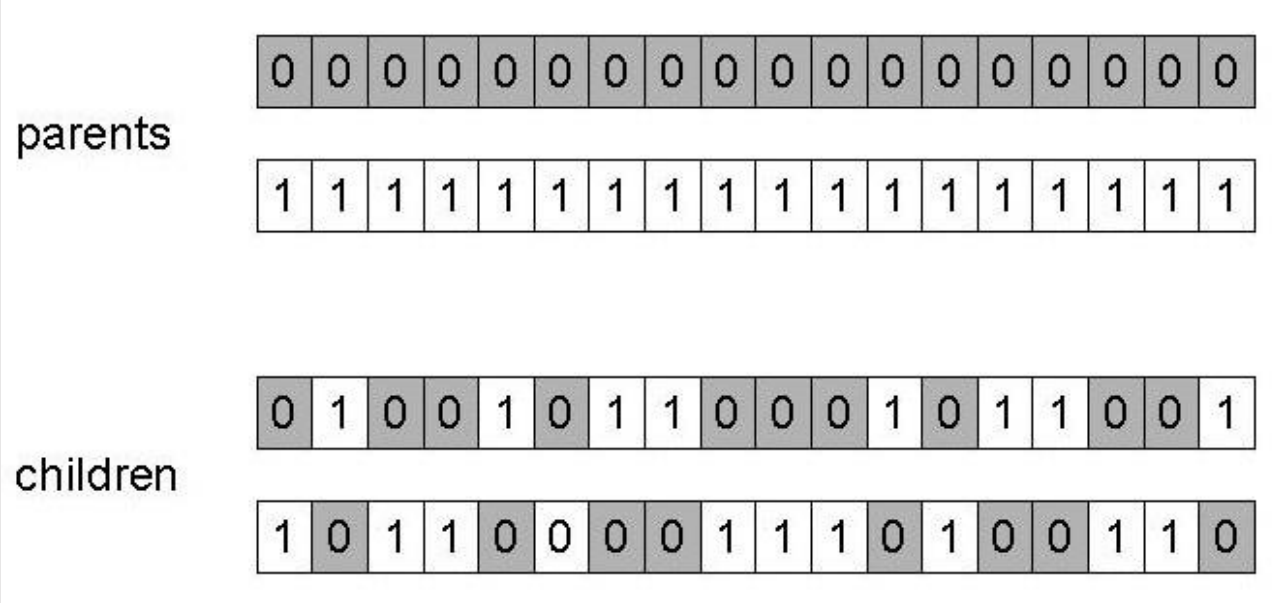
Cruza de n-puntos

- Selecciona aleatoriamente n puntos de cruza
- Se divide cromosoma en estos puntos
- Alterna entre los dos padres para generar los descendientes



Cruza uniforme

- Se recorre uno a uno los elementos del cromosoma y se elige si se selecciona la información de uno u otro padre (como si se tratará de un volado “flip”)



Operador: Mutación

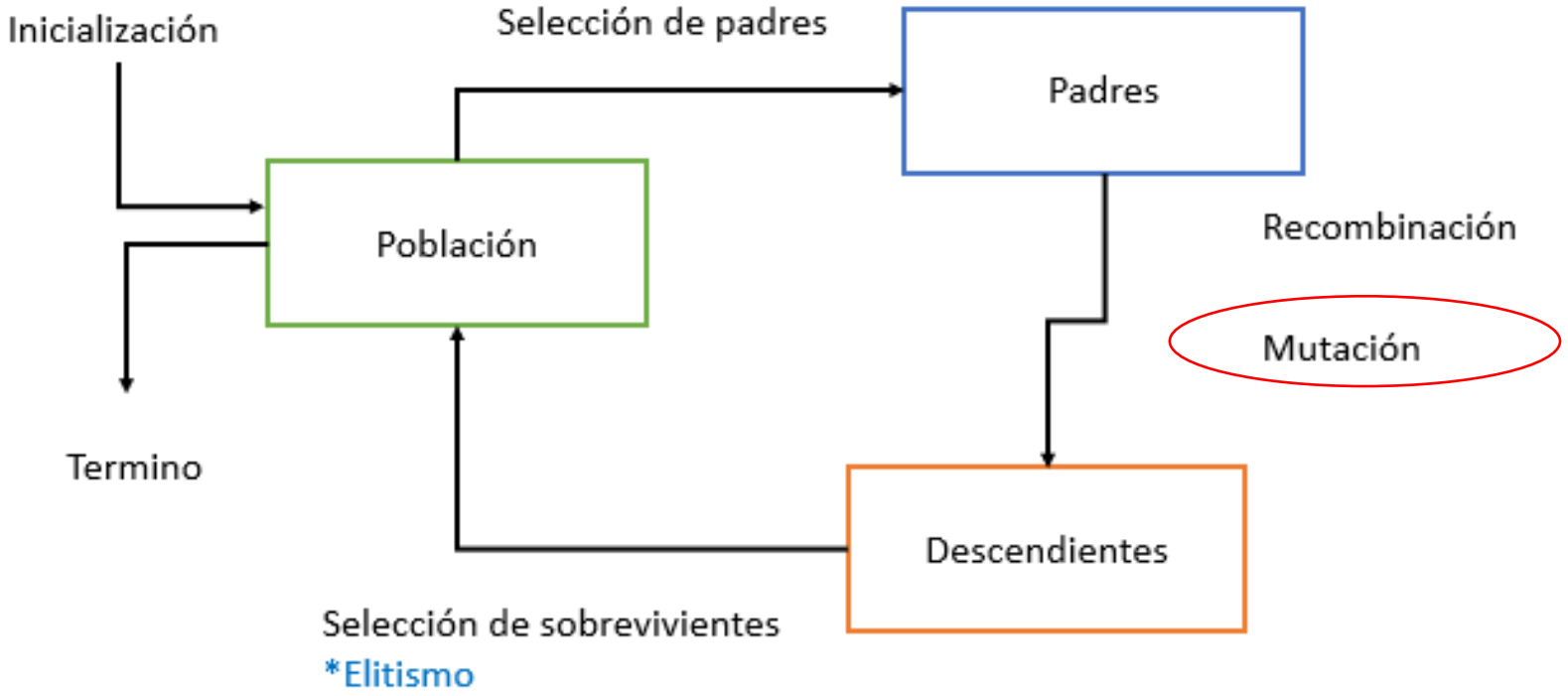


Figura 2.2. Esquema General de un algoritmo Evolutivo como un diagrama de flujo

Mutación Binaria

❑Antes de aplicar mutación:

s_1`` = 1110110101

s_2`` = 1111010101

s_3`` = 1110111101

s_4`` = 0111000101

s_5`` = 0100011101

s_6`` = 1110110011

❑ Después de aplicar mutación :

s_1``` = 1110100101

s_2``` = 1111110100

s_3``` = 1110101111

s_4``` = 0111000101

s_5``` = 0100011101

s_6``` = 1110110001

Operador de cruza VS mutación

- Durante decadas se ha discutido cuál de los dos operadores es mejor
- La respuesta considera lo siguiente:
  - Esto depende del problema,
  - Sin embargo en general es major considerer ambos operadores

References

- *Genetic Algorithms: A Tutorial* By Dr. Nysret Musliu , Associate Professor Database and Artificial Intelligence Group, Vienna University of Technology.
- Introduction to Genetic Algorithms, Assaf Zaritsky Ben-Gurion University, Israel (www.cs.bgu.ac.il/~assafza)