



# Inteligencia Artificial

**Unidad 1:** Representación Avanzada del Conocimiento

**TEMA 2:** Algoritmos de la IA Clásica

**Módulo 2:** Algoritmos Genéticos

Unidad 1

# Representación Avanzada del Conocimiento

**TEMA 2:** Algoritmos de la IA Clásica

Sesión 7

## MÓDULO 2: Algoritmos Genéticos



### Contenido

1. Origen del Algoritmo Genético
2. ¿Cómo funciona el Algoritmo Genético?
3. Fases en un Algoritmo Genético
4. Ventajas y limitaciones del Algoritmo Genético
5. Ejemplos de aplicación

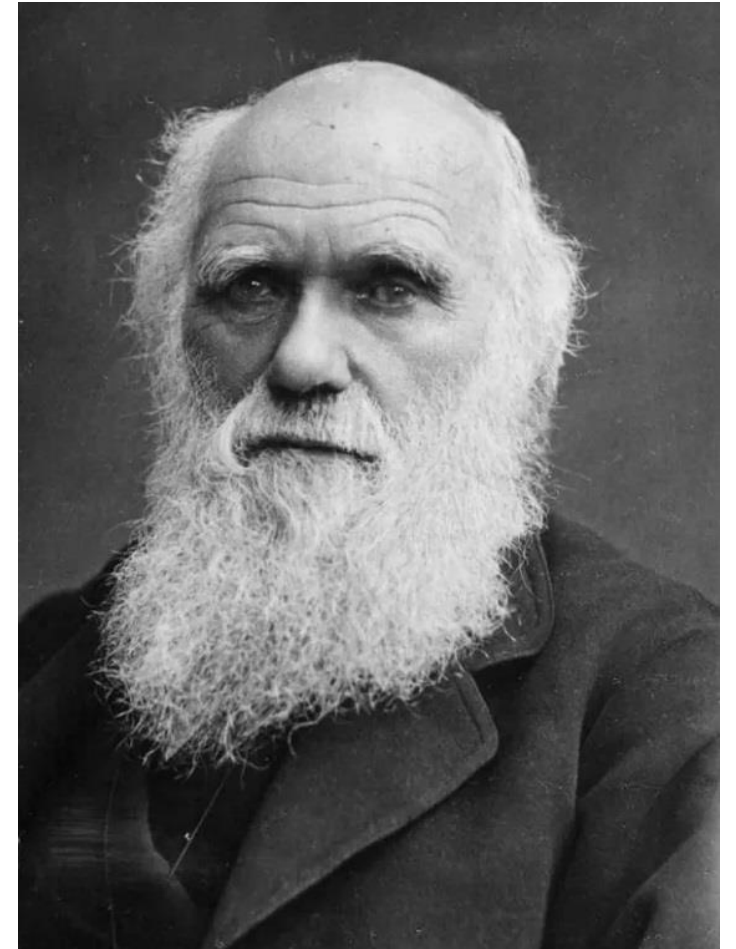


### Preguntas

# 1. Origen del Algoritmo Genético en I.A.

Según **Charles Darwin**, el conocido naturalista inglés, describía en **El origen de las especies**:

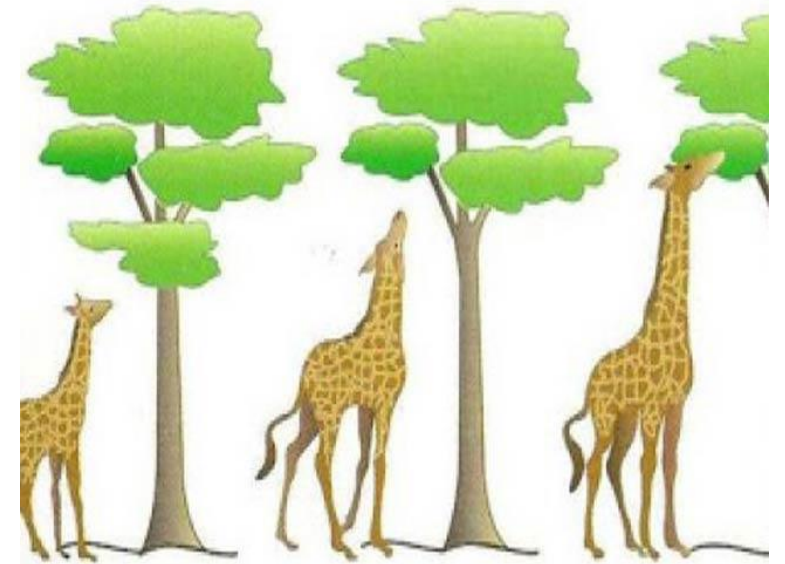
“La capacidad biológica que tienen los seres vivos de adaptarse mediante la conocida como selección natural, en donde no sobrevivía aquel ser vivo más fuerte o más inteligente, sino quien logre adaptarse mejor”.



# 1. Origen del Algoritmo Genético en I.A.

El **Algoritmo Genético en I.A.** es un algoritmo que se inspira en la teoría de la selección natural de Darwin para **resolver problemas de optimización**.

Este algoritmo refleja el proceso de selección natural en el que se seleccionan los individuos más aptos para la reproducción con el fin de producir descendencia de la próxima generación.



## 2. ¿Cómo funciona el Algoritmo Genético?

### Noción de selección natural

- ❑ El proceso de selección natural comienza con la selección de los individuos más aptos de una población.
- ❑ Producen descendientes que heredan las características de los padres y se agregarán a la próxima generación.
- ❑ Si los padres tienen una mejor condición física, sus hijos serán mejores que los padres y tendrán más posibilidades de sobrevivir.
- ❑ Este proceso sigue iterando y al final, se encontrará una generación con las personas más aptas.

**ESTA NOCION ES UNA  
HEURÍSTICA QUE LA  
APLICAMOS PARA  
RESOLVER PROBLEMAS DE  
BUSQUEDA**



Donde consideramos un conjunto de soluciones para un problema y seleccionamos el conjunto de las mejores de entre ellas.

### 3. Fases en un Algoritmo Genético

- La secuencia de fases de este algoritmo se repiten para producir individuos en cada nueva generación que son mejores que la generación anterior.
- La población tiene un tamaño fijo. A medida que se forman las nuevas generaciones, mueren los individuos con menos aptitud física, lo que proporciona espacio para nuevas crías.

¿Cuáles son las **FASES** que componen el Algoritmo Genético?



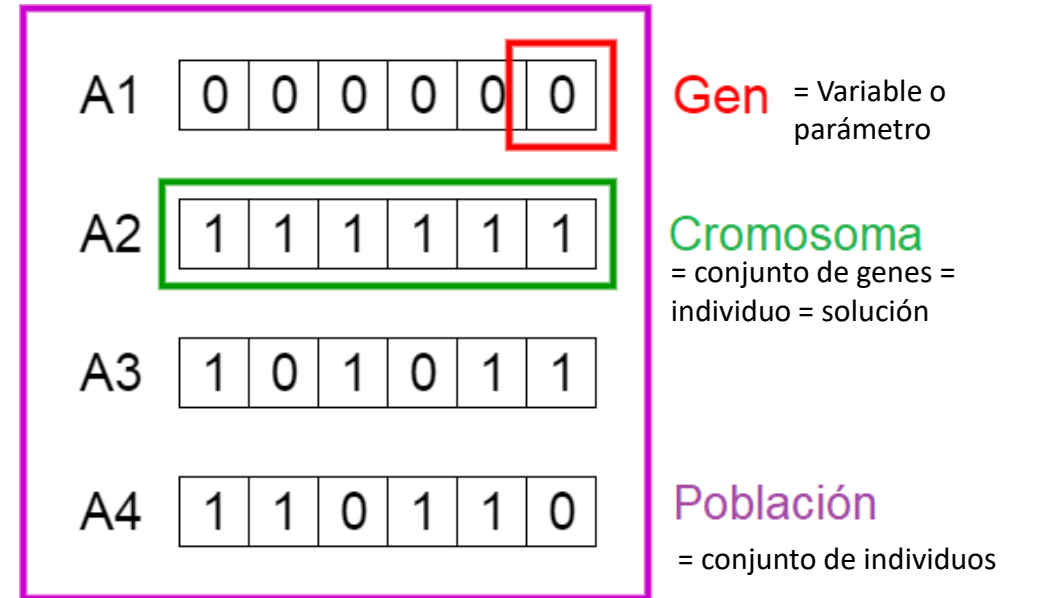
Se consideran cinco fases en un algoritmo genético:

1. Población inicial
2. Función fitness
3. Selección
4. Cruce
5. Mutación

# 3. Fases en un Algoritmo Genético

## 1. POBLACIÓN INICIAL

- ❑ El proceso comienza con un conjunto de individuos que se denomina **Población**. Cada individuo es una solución al problema que se desea resolver (inicialmente establecidos o elegidos de forma aleatoria).
- ❑ Un individuo se caracteriza por un conjunto de parámetros (variables) conocidos como **genes**. Los genes se unen en una cuerda para formar un cromosoma (solución).
- ❑ Los valores de los cromosomas se representan en binario como una secuencia de 0 y 1 (aunque otra codificación también es posible).



individuo

A1, A2, A3 y A4 (son los estados o nodos)

Cada estado o nodo es una solución al problema

# 3. Fases en un Algoritmo Genético

## 2. FUNCIÓN DE FITNESS

- ❑ La función de aptitud o fitness determina qué tan en forma está un individuo (la capacidad de un individuo para competir con otros individuos).
- ❑ Da una puntuación de aptitud física a cada individuo.
- ❑ La probabilidad de que un individuo sea seleccionado para reproducción se basa en su puntuación de aptitud.

A1	0010	0101	1100	1000	1010	10
A2	0110	0001	1101	0110	1111	14
A3	0010	0111	1000	0011	1011	4
A4	0000	1101	1101	0100	1110	1

Valores de aptitud o fitness

Más aptitud ➡ Mejor solución ➡ Mayor probabilidad de sobrevivir...



# 3. Fases en un Algoritmo Genético

## 3. SELECCION

- ❑ La idea de la fase de selección es seleccionar a los individuos más aptos y dejarlos pasar sus genes a la siguiente generación.
- ❑ Se seleccionan dos pares de individuos (padres) en función de sus puntuaciones de aptitud física.
- ❑ Los individuos con una alta aptitud física tienen más posibilidades de ser seleccionados para la reproducción.

- En cada generación, los valores de aptitud física se calcula para cada cromosoma (individuo).

A1	0010	0101	1100	1000	1010	20
A2	0110	0001	1101	0110	1111	14
A3	0010	0111	1000	0011	1011	13
A4	0000	0111	1101	0110	1110	8
A5	0010	1111	1000	0111	1010	4
A6	0000	1101	1101	0101	1110	1

- La mayoría de los individuos aptos sobreviven para reproducirse.
- Los individuos de baja aptitud física se eliminan de la población.



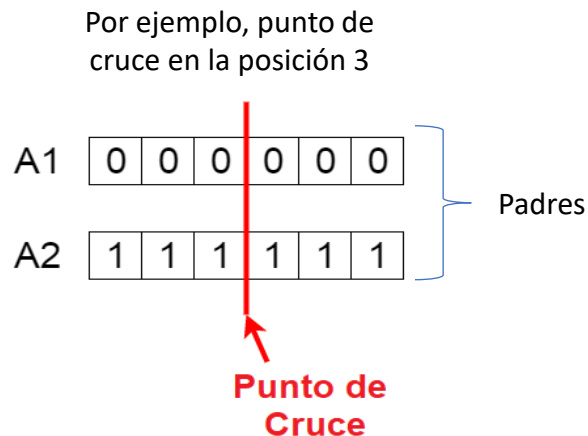
**Sobrevive el más apto**

### 3. Fases en un Algoritmo Genético

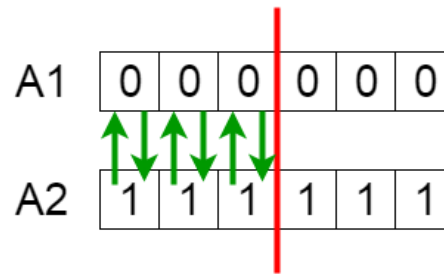
#### 4. CRUCES (Crossover)

El cruce es la fase más significativa de un algoritmo genético.

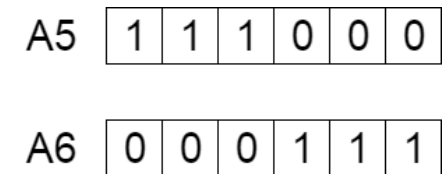
- ❑ Para cada par de padres, en el acto de aparearse, se elige un **punto de cruce** al azar dentro de los genes.



- ❑ **La descendencia** se crea intercambiando los genes de los padres entre ellos hasta que se alcanza el punto de cruce.

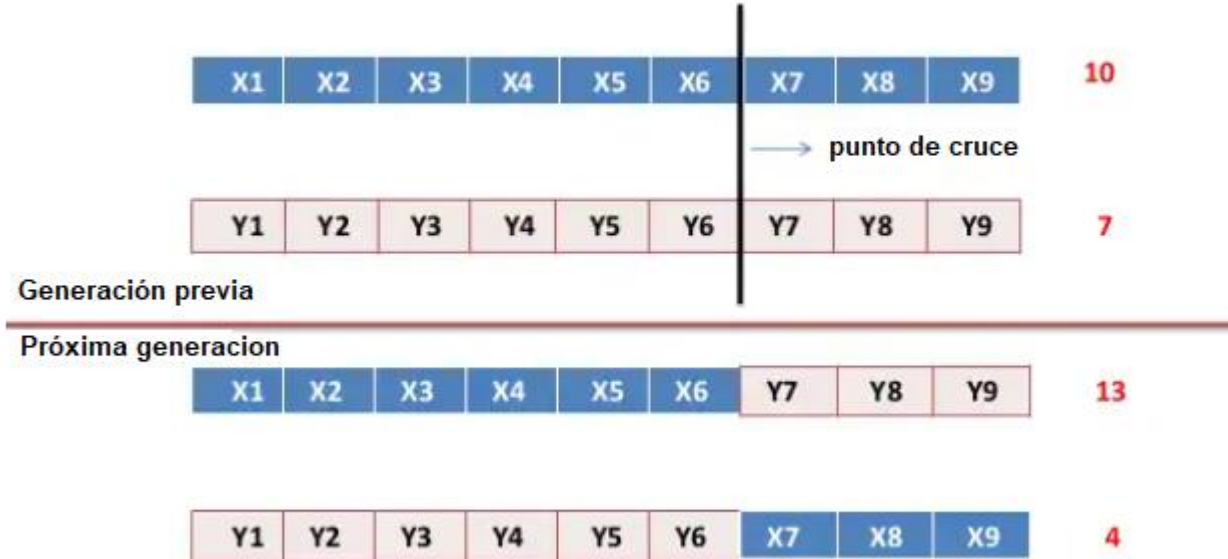


- ❑ La nueva descendencia se agrega a la población.



### 3. Fases en un Algoritmo Genético

#### CRUCE



# 3. Fases en un Algoritmo Genético

## 4. MUTACION

La mutación se realiza para mantener la diversidad genética (diversidad dentro de la población).

- ☐ La mutación puede ocurrir:
  - En uno más o más lugares en el cromosoma.
  - En muchos cromosomas en una generación.
- ☐ Es probabilista: en una nueva generación, algunos de sus genes pueden estar sujetos a una mutación con una probabilidad aleatoria baja.
- ☐ La mutación implica que algunos de los genes de la cadena de genes (cromosoma) se pueden invertir.

Antes de la Mutación

A5 

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

Después de la Mutación

A5 

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

### 3. Fases en un Algoritmo Genético

- ❑ El algoritmo termina si la población ha convergido (no produce descendencia que sea significativamente diferente de la generación anterior).
- ❑ Entonces, decimos que el algoritmo genético ha aportado un conjunto de soluciones a nuestro problema.

¿Cuándo termina la ejecución el Algoritmo Genético?



#### ALGORITMO GENÉTICO

1. Elija la población inicial.
2. Evalúe la aptitud de cada individuo.
3. Repita hasta el límite de tiempo, aptitud suficiente lograda, numero de iteraciones (generaciones), etc. a la descendencia:
  - Seleccione los individuos más aptos para la reproducción.
  - Cree nuevos individuos mediante operaciones de cruzamiento y mutación para dar a luz a la descendencia (opera sobre dos cromosomas a la vez y genera dos descendientes).
  - Evalúe la aptitud individual de nuevos individuos.
  - Reemplace la población menos apta por nuevos individuos más aptos.

Fuente:

[www.eis.uva.es/elena/newcomersGAs.htm](http://www.eis.uva.es/elena/newcomersGAs.htm)

# 3. Fases en un Algoritmo Genético

## 1. Fase de Inicialización

Selección aleatoria de la población



## 2 Fase de Evaluación

A cada individuo (cromosoma) se evalúa su aptitud.



Mejor solución  $X^*$

## 3. Fase de Selección

Conocida la aptitud de cada cromosoma se procede a elegir los cromosomas que serán cruzados en la siguiente generación (interacción).



## 4. Cruce

O recombinación, representa la reproducción sexual (principal operador), opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos cromosomas padres.



## 6. Reemplazo (actualización de generación)

Una vez aplicados los operadores genéticos (cruce), se seleccionan los mejores individuos para conformar la nueva población de la generación siguiente.



## 5. Mutación

Modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual.

# 4. Ventajas y limitaciones del Algoritmo Genético

## VENTAJAS

- **Son algoritmos robustos.**  
Ampliamente probados y con resultados de optimización de calidad.
- **Son Paralelos:** tienen descendencia múltiple, pueden explorar el espacio de soluciones en múltiples direcciones a la vez.  
A diferencia de los algoritmos anteriores que son en serie y sólo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo, y si la solución que descubren resulta subóptima, no se puede hacer otra cosa que abandonar todo el trabajo hecho y empezar de nuevo.
- **Proporcionan optimización sobre espacios de estado grandes.**  
Debido al paralelismo que les permite evaluar implícitamente muchos esquemas a la vez.  
Ha demostrado su efectividad al evitar muchos óptimos locales y descubrir el óptimo global incluso en espacios grandes y complejos.

## LIMITACIONES

(Aunque todas ellas pueden superarse y que ninguna afecta a la validez de la evolución biológica)

- **Mala definición de la representación del problema.**  
El lenguaje utilizado para especificar soluciones candidatas debe ser robusto; es decir, debe ser capaz de tolerar cambios aleatorios que no produzcan constantemente errores fatales o resultados sin sentido (por ello suele representarse a los individuos como cadenas binarias).
- **Escribir erradamente la función de aptitud.**  
Si se elige mal una función de aptitud o se define de manera inexacta, puede que el algoritmo genético sea incapaz de encontrar una solución al problema, o puede acabar resolviendo el problema equivocado.
- **La convergencia prematura.**  
Si un individuo que es más apto que la mayoría de sus competidores emerge muy pronto en el curso de la ejecución, se puede reproducir tan abundantemente que merme la diversidad de la población demasiado pronto, provocando que el algoritmo converja hacia el óptimo local que representa ese individuo (sesgo). Esto es un problema especialmente común en las poblaciones pequeñas.

# 5. Ejemplos aplicando el Algoritmo Genético

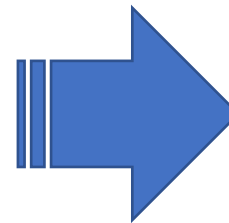
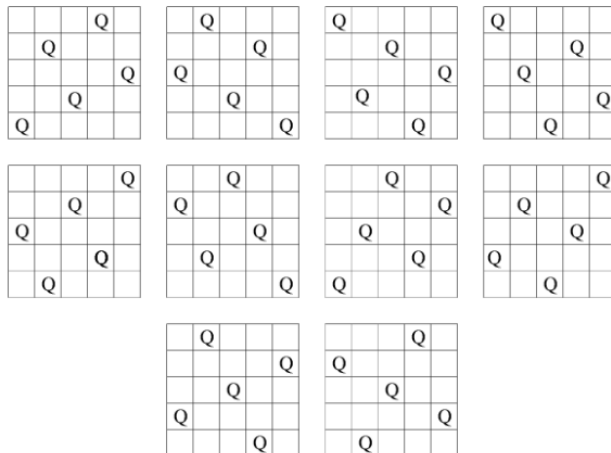
## EJEMPLO #1: el problema de N-Queen (N-reinas)

**Descripción del problema:** En este problema, necesitamos encontrar una disposición de N reinas en el tablero de ajedrez, de modo que ninguna reina pueda atacar a otras reinas en el tablero (ni de forma vertical, horizontal o diagonal).

**Solución a través de Algoritmo Genético:** usar el principio de evolución para encontrar una solución a un problema.

**Estados:** 5 Reinas      **Espacio:** Tablero de 5x5

### Visualización de todos los casos posibles



### Pasos a seguir:

1. Diseño de cromosomas
2. Inicialización
3. Evaluación de aptitud
4. Selección
5. Crossover
6. Mutación
7. Actualización de generación
8. Regresar al paso 3.

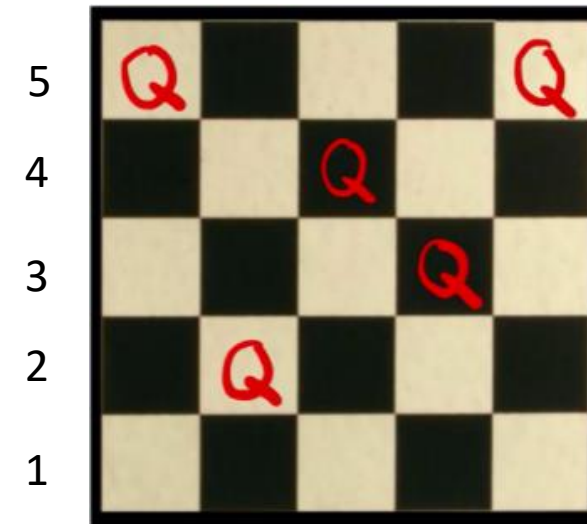


# 5. Ejemplos aplicando el Algoritmo Genético

## SOLUCION AL PROLEMA DE N-Queen (ejemplo para 5-reinas)

### 1. Diseño de cromosomas

- ☐ Creamos una representación cromosómica.
- ☐ Para mostrar un cromosoma, la mejor manera es representarlo como una lista de longitud **N** donde en nuestro caso **N = 5**.
- ☐ El valor de cada índice muestra la fila de la reina en una columna. El valor de cada índice es de 1 a 5.



CROMOSOMA =  
(LISTA DE LONGITUD 5)

5	2	4	3	5
---	---	---	---	---

# 5. Ejemplos aplicando el Algoritmo Genético

## 2. Inicialización

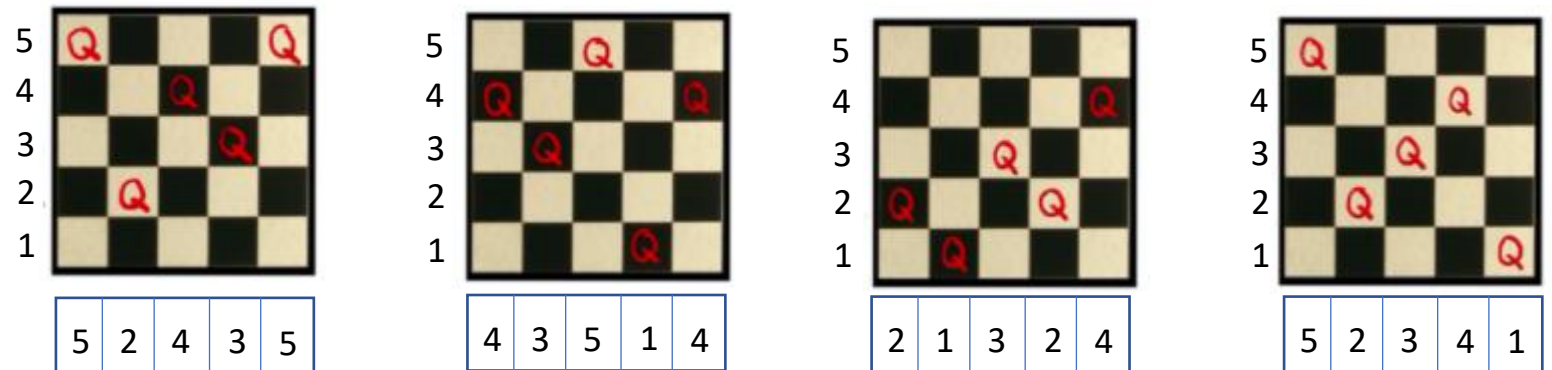
- ❑ Necesitamos crear y organizar una población aleatoria de cromosomas (posibles soluciones) .
- ❑ Por ejemplo, tomamos 4 cromosomas (individuos), cada uno de los cuales tiene una longitud de 5.

- ❑ Los 4 cromosomas son:

[5 2 4 3 5]	<table><tr><td>5</td><td>2</td><td>4</td><td>3</td><td>5</td></tr></table>	5	2	4	3	5
5	2	4	3	5		
[4 3 5 1 4]	<table><tr><td>4</td><td>3</td><td>5</td><td>1</td><td>4</td></tr></table>	4	3	5	1	4
4	3	5	1	4		
[2 1 3 2 4]	<table><tr><td>2</td><td>1</td><td>3</td><td>2</td><td>4</td></tr></table>	2	1	3	2	4
2	1	3	2	4		
[5 2 3 4 1]	<table><tr><td>5</td><td>2</td><td>3</td><td>4</td><td>1</td></tr></table>	5	2	3	4	1
5	2	3	4	1		

Cada cromosoma tiene 5 genes

- ❑ Cada uno puede mostrarse en el tablero de ajedrez de la siguiente forma:



Población = 4 = cromosomas = individuos

## 5. Ejemplos aplicando el Algoritmo Genético

### 3. Evaluación de aptitud

- ❑ La función de aptitud **son pares de reinas que no atacan**, por lo tanto, las puntuaciones más altas son mejores.
- ❑ Para resolver la función de aptitud del cromosoma [5 2 4 3 5], asignamos a cada reina de forma única Q1, Q2, Q3, Q4 y Q5.
- ❑ Para encontrar el valor de la función de aptitud utilizamos la siguiente ecuación:

$$\text{Función fitness} = F1 + F2 + F3 + F4 + F5$$



**Cálculo de la función de aptitud del cromosoma [5 2 4 3 5]**

Donde:

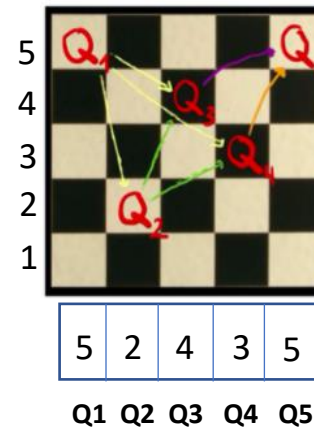
**F1** = número de parejas de reinas que no atacan con la reina Q1.

**F2** = número de parejas de reinas que no atacan con la reina Q2.

**F3** = número de parejas de reinas que no atacan con la reina Q3.

**F4** = número de parejas de reinas que no atacan con la reina Q4.

**F5** = número de parejas de reinas que no atacan con la reina Q5.



$$\text{Función fitness} = F1 + F2 + F3 + F4 + F5$$

$$\text{Función fitness} = 3 + 2 + 1 + 1 + 0$$

$$\text{Función fitness} = 7$$

Con el mismo procedimiento, debemos evaluar todos los individuos de nuestra población (cromosomas) usando la función de aptitud.

## 5. Ejemplos aplicando el Algoritmo Genético

### 3. Evaluación de aptitud - Continuación (1)

- ❑ Calculamos la función de para los cromosomas faltantes:

5	2	4	3	5
4	3	5	1	4
2	1	3	2	4
5	2	3	4	1

**Función fitness = 7**  
(calculada en diapositiva anterior)

**Función fitness = 6**

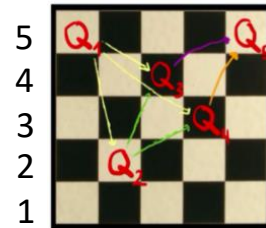
**Función fitness = 6**

**Función fitness = 4**

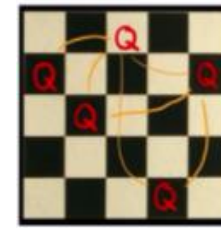


### Representación de estados y cálculo de la función fitness

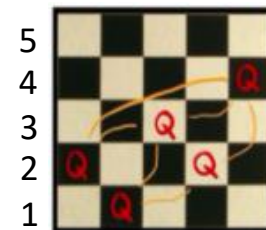
**Función fitness =  $F1 + F2 + F3 + F4 + F5$**



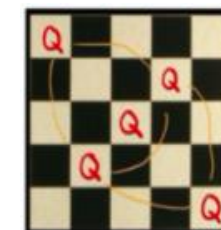
**Función fitness = 7**



**Función fitness = 6**



**Función fitness = 6**



**Función fitness = 4**

5	2	4	3	5
---	---	---	---	---

**7**

4	3	5	1	4
---	---	---	---	---

**6**

**Función fitness =  $(1+2+2+1+0)$**

2	1	3	2	4
---	---	---	---	---

**6**

**Función fitness =  $(2+2+1+1+0)$**

5	2	3	4	1
---	---	---	---	---

**4**

**Función fitness =  $(2+1+0+1+0)$**

## 5. Ejemplos aplicando el Algoritmo Genético

### 3. Evaluación de aptitud - Continuación (2)

- ❑ Calculamos la probabilidad de ser elegido de la función de aptitud (necesario para el siguiente paso de selección).
- ❑ Sumamos todas las funciones de aptitud obtenidas de los cromosomas evaluados:  $7 + 6 + 6 + 4 = 23$
- ❑ Dividimos la función de aptitud de cada cromosoma (individuo) entre el total obtenido de la función de aptitud de la población y la multiplicamos por 100%.

5	2	4	3	5
---	---	---	---	---

**Función fitness =** 7

probabilidad de ser elegido =  $7/23 * 100\% = 30.4\%$

4	3	5	1	4
---	---	---	---	---

**Función fitness =** 6

probabilidad de ser elegido =  $6/23 * 100\% = 26.1\%$

2	1	3	2	4
---	---	---	---	---

**Función fitness =** 6

probabilidad de ser elegido =  $6/23 * 100\% = 26.1\%$

5	2	3	4	1
---	---	---	---	---

**Función fitness =** 4

probabilidad de ser elegido =  $4/23 * 100\% = 17.4\%$

# 5. Ejemplos aplicando el Algoritmo Genético

## 4. Selección

- ❑ Elegimos al azar los dos pares para reproducir en función de las probabilidades que calculamos en el paso anterior (una cierta cantidad de cromosomas sobrevivirá en el próximo generador usando un operador de selección).
- ❑ Tomamos aleatoriamente los siguientes cromosomas en función de sus probabilidades:

5	2	4	3	5	probabilidad de ser elegido = 30.4%	✓
4	3	5	1	4	probabilidad de ser elegido = 26.1%	✓
2	1	3	2	4	probabilidad de ser elegido = 26.1%	✓
5	2	3	4	1	probabilidad de ser elegido = 17.4%	✓

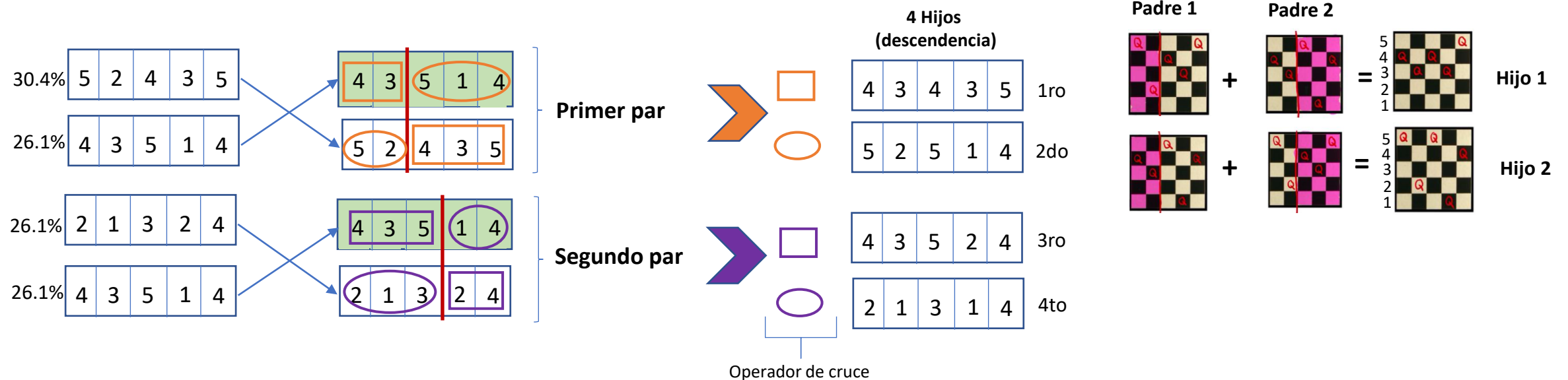
- ❑ Podemos notar que no tomamos el cromosoma [5 2 3 4 1] porque su probabilidad de ser elegido es la menor entre los cromosomas (17.4%).



# 5. Ejemplos aplicando el Algoritmo Genético

## 5. Cruce (Crossover)

- ❑ Para el primer par [4 3 5 1 4] y [5 2 4 3 5] el punto de cruce se seleccionará después de dos genes.
- ❑ En el caso del segundo par [4 3 5 1 4] y [2 1 3 2 4] el punto de cruce se elegirá después de tres genes.



- ❑ En el cruce, los cromosomas seleccionados actúan como padres que se combinan mediante el operador de cruce para crear hijos. En otras palabras, combina la información genética de dos padres para generar nueva descendencia.

## 5. Ejemplos aplicando el Algoritmo Genético

### 6. Mutación

- ❑ En el proceso de mutación, **alteramos uno o más valores de genes** en los cromosomas que encontramos después del cruce (en la descendencia).
- ❑ Entonces cambiaremos aleatoriamente algunos genes y la probabilidad de mutación es baja. Entonces, en nuestro ejemplo, nuestra mutación se verá así:

[4 3 4 3 5] → [4 3 **1** 3 5]

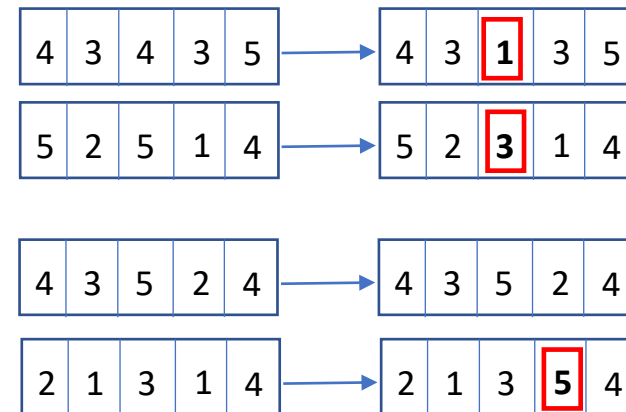
[5 2 5 1 4] → [5 2 **3** 1 4]

[4 3 5 2 4] → [4 3 5 2 4]

[2 1 3 1 4] → [2 1 3 **5** 4]



#### Mutación



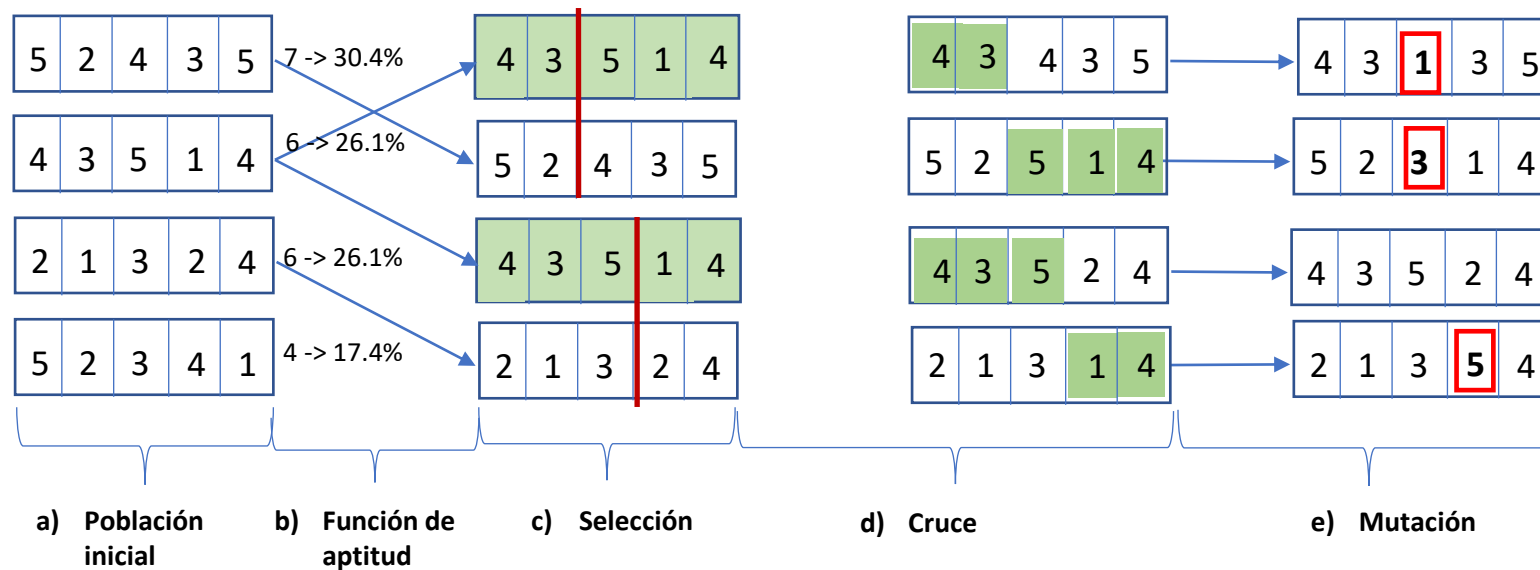
- ❑ Podemos notar que el tercer gen en el cromosoma [4 3 4 3 5] cambió de 4 a 1.
- ❑ El tercer gen en el cromosoma [5 2 5 1 4] cambió de 5 a 3.
- ❑ Además de esto, el cuarto gen en el cromosoma [2 1 3 1 4] cambió de 1 a 5.



# 5. Ejemplos aplicando el Algoritmo Genético

## 7. Actualización de generación

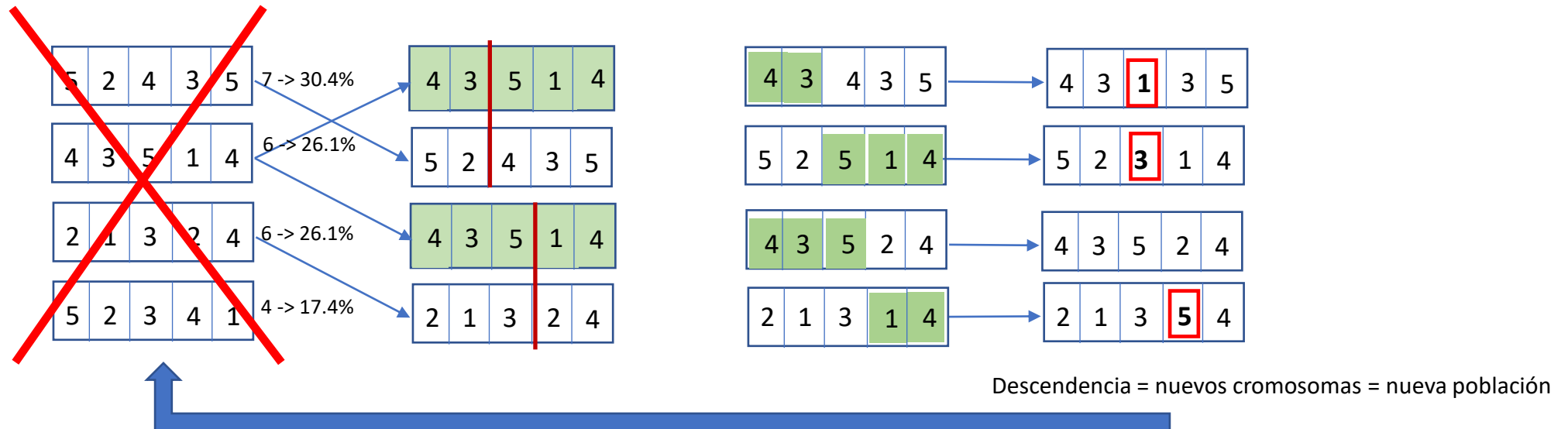
Antes de proceder con la actualización de la generación, veamos hasta donde el algoritmo genético ha llegado para resolver el algoritmo de 5 reinas:



# 5. Ejemplos aplicando el Algoritmo Genético

## 7. Actualización de generación

- ❑ Ahora si, en este paso, necesitamos actualizar la generación. Los nuevos cromosomas actualizarán la población, pero el número de individuos (cromosomas) de la población no cambiará.
- ❑ Entonces los cromosomas (descendientes) serán nuestra nueva población: [4 3 1 3 5] [5 2 3 1 4] [4 3 5 2 4] [2 1 3 5 4]



- ❑ Entonces, en el siguiente paso, debemos volver al paso 3 (evaluación de la aptitud) para encontrar la función de aptitud de nuestra población actualizada.

## 5. Ejemplos aplicando el Algoritmo Genético

### 8. Regresar al paso 3 → Evaluación de aptitud

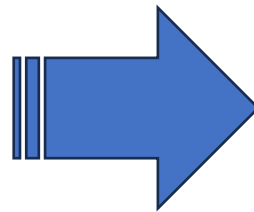
- ❑ Los pasos 3 a 7 se repiten hasta que el cromosoma (solución) satisfaga lo siguiente: **Valor de aptitud = Fmax**, donde Fmax es igual a 10
- ❑ En la siguiente iteración los individuos de integran la población deben ir mejorando u obteniendo un mayor resultado en la función de aptitud

4	3	1	3	5	<b>Función fitness =</b>	8	probabilidad de ser elegido = $8/31 * 100\% =$	25.8%
5	2	3	1	4	<b>Función fitness =</b>	8	probabilidad de ser elegido = $8/31 * 100\% =$	25.8%
4	3	5	2	4	<b>Función fitness =</b>	8	probabilidad de ser elegido = $8/31 * 100\% =$	25.8%
2	1	3	5	4	<b>Función fitness =</b>	7	probabilidad de ser elegido = $7/31 * 100\% =$	22.50%
Nueva población o generación					3. Evaluación de aptitud o fitness			

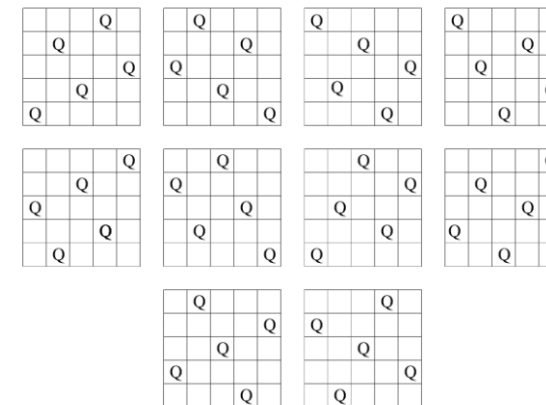
## 5. Ejemplos aplicando el Algoritmo Genético

### 8. Regresar al paso 3 → Evaluación de aptitud

- ❑ Al evaluar el resultado de la función de aptitud, se elegirá a los mejores descendientes que luego se cruzarán y generaran mejores individuos que reemplazaran a la población anterior.
- ❑ Este proceso llevado a cabo en cada iteración, debe llevar a alcanzar el mejor valor (valor máximo de la función de aptitud), que es aquel con el cual se alcanza la solución al problema.
- ❑ La **función de aptitud** es la que guía al Algoritmo Genético (AG) para encontrar las soluciones más prometedoras, al calificar la población generada a fin de encontrar la solución al problema (estado final o estado objetivo).
- ❑ Resulta de mucha importancia establecer correctamente el valor máximo o correcto que debe alcanzar la función de aptitud, porque de ella dependerá el alcanzar la mejor solución al problema.
- ❑ En el ejemplo, la función de aptitud debe llegar al valor máximo **FMAX = 10** que significa que ninguna reina puede atacar a otra en el tablero.



#### Estados Objetivo



Cada estado  
objetivo obtiene  
el valor  
**FMAX = 10**

# 5. Ejemplos aplicando el Algoritmo Genético

## EJEMPLO #2: Población binaria

**Descripción del problema:** Dado un conjunto de 5 genes, cada gen puede contener uno de los valores binarios 0 y 1.

**Solución a través de Algoritmo Genético:** usar el principio de evolución para encontrar una solución a un problema.

### Función de aptitud (fitness)

El valor de aptitud se calcula como el número de 1 presentes en el genoma.

Si hay cinco 1, entonces está teniendo la máxima aptitud. Si no hay 1, entonces tiene la aptitud mínima.

### Objetivo

Este problema a resolver con el algoritmo genético intenta maximizar la función de aptitud para proporcionar una población formada por el individuo más apto, es decir, individuos con cinco 1.

Descargar y ejecutar este ejemplo codificado en Java: <https://github.com/memento/GeneticAlgorithm>

## 5. Ejemplos aplicando el Algoritmo Genético

**EJEMPLO #3:** Comparación de cadenas de texto.

**Descripción del problema:** Crear una cadena de destino, comenzando desde una cadena aleatoria usando algoritmo genético

**Solución a través de Algoritmo Genético:** usar el principio de evolución para encontrar una solución a un problema.

**Función de aptitud (fitness)**

El valor de aptitud se calcula como el número de coincidencias entre la cadena destino y una cadena aleatoria.

Si hay coincidencia entre ambas cadenas, entonces está teniendo la máxima aptitud. Si no hay, entonces se obtiene la mejor solución.

**Descargar y ejecutar este ejemplo codificado en Python desde el Aula Virtual**

# PREGUNTAS

Dudas y opiniones