



Universidad  
Andrés Bello

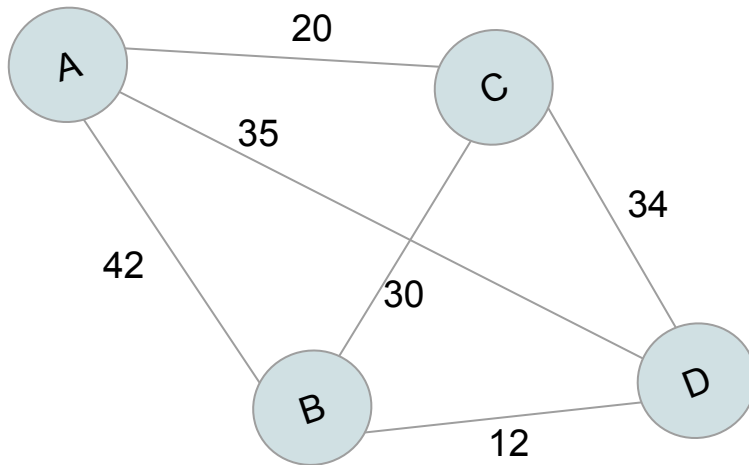
# Algoritmos Genéticos

...

Jean Contreras  
Sistemas Inteligentes

# Problema del Vendedor Viajero

Recorrer todas las ciudades en la menor distancia posible, pasando solo una vez por cada una:



**Solución Factible:** A -> B -> C -> D -> A

$$42 \rightarrow 30 \rightarrow 34 \rightarrow 35 = 141$$

**Solución Óptima:** A -> C -> B -> D -> A

$$20 \rightarrow 12 \rightarrow 34 \rightarrow 35 = 97$$

Es un tipo de algoritmo de búsqueda de soluciones, basado en la naturaleza, aplicando conceptos tales como:

- Individuo
- Fitness
- Gen
- Cromosoma
- Población inicial
- Cruzamiento
- Recombinación
- Mutación

Un individuo, para el caso del vendedor viajero corresponde a una solución factible del problema.

**SF1:** A  $\rightarrow$  B  $\rightarrow$  C  $\rightarrow$  D  $\rightarrow$  A

Fitness corresponde a la calidad de la solución, que en este caso es la distancia total recorrida.

**FITNESS (SF1):** 42  $\rightarrow$  30  $\rightarrow$  34  $\rightarrow$  35 = **141**

A menor Fitness, la solución está más cerca del óptimo.

# Función de Fitness

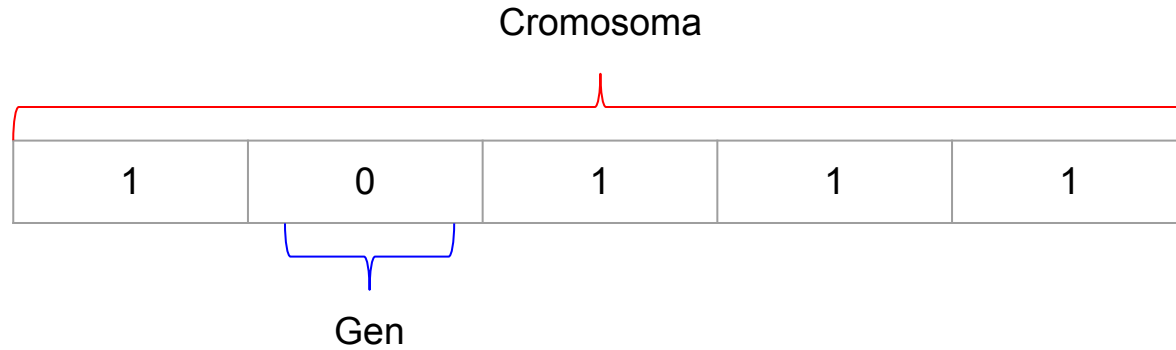
Dependiendo de la problemática a resolver, se puede definir una función de Fitness u otra.

Para el problema del Vendedor Viajero usamos simplemente el coste del trayecto, pero hay alternativas:

$$Fitness(X) = \frac{1}{cost(x)}$$

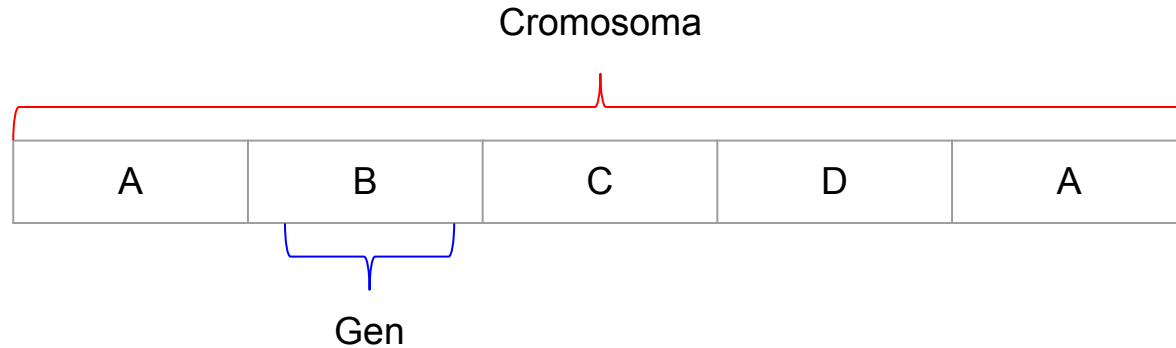
# Cromosoma

Es un conjunto de genes de un individuo (solución) representados en una lista:



# Cromosoma

Es un conjunto de genes de un individuo (solución) representados en una lista:



# Población inicial

Conjunto de individuos al azar, utilizados como base para la obtención de una mejor solución.

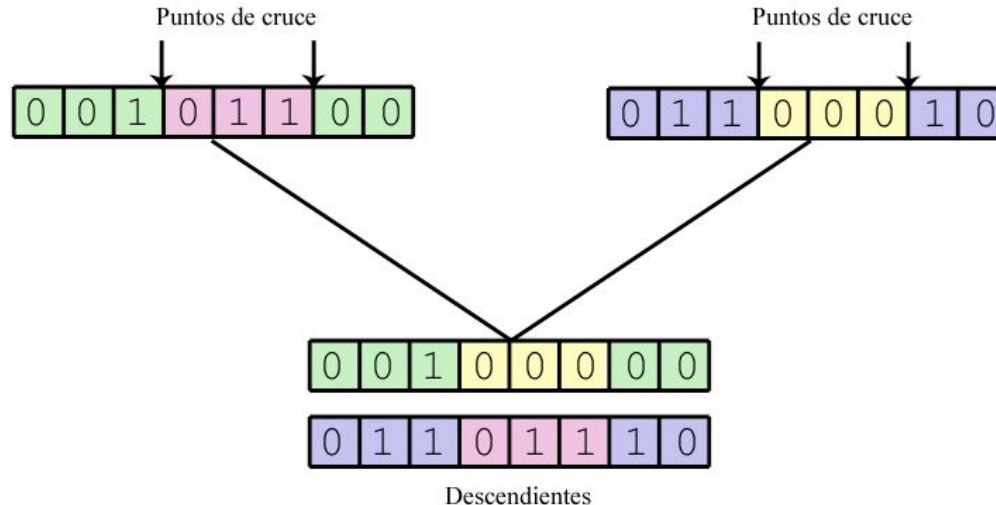
<b>SF1</b>	A	B	C	D	A	<b>FITNESS (SF1): 0.0071</b>
<b>SF2</b>	A	C	B	D	A	<b>FITNESS (SF1): 0.0103</b>
<b>SF3</b>	A	B	D	C	A	<b>FITNESS (SF1): 0.0093</b>

Se recomienda una población comprendida entre 1 y 21 individuos.



# Cruzamiento

Se eligen dos individuos (dependiendo de su fitness), luego se selecciona un conjunto de genes de cada uno (puntos de cruce), generando dos descendientes al combinar sus características, cuya calidad puede ser mejor que la de sus padres.



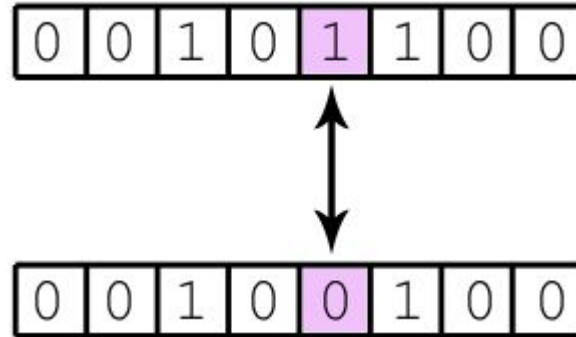
# Recombinación

Eliminar soluciones malas o de baja calidad, para dar énfasis a las buenas soluciones.

<b>SF2</b>	A	B	C	D	A	<b>FITNESS (SF2): 0.0071</b>
<b>SF3</b>	A	C	B	D	A	<b>FITNESS (SF3): 0.0103</b>
<b>SF4</b>	A	B	D	C	A	<b>FITNESS (SF4): 0.0093</b>

# Mutación

Cambio aleatorio de un gen de un individuo, puede mejorar o empeorar la solución.



**Elitismo:** Para que una buena solución no se pierda al mutar, lo ideal es realizar una copia del individuo a la nueva población.

# Término del Algoritmo

Los Algoritmos Genéticos poseen una condición de término, esta puede ser un número máximo de generaciones, tiempo límite, etc.

Al terminar el Algoritmo, se elige el individuo de mayor calidad como la solución al problema.

Puede que la solución no sea el óptimo, pero se encuentra cercano a este.



**SF2**

A	C	B	D	A
---	---	---	---	---

**FITNESS (SF1): 0.0103**

# Algoritmo Genético Simple



Universidad  
Andrés Bello

```
BEGIN /* Algoritmo Genetico Simple */
  Generar una poblacion inicial.
  Computar la funcion de evaluacion de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generacion */
      FOR Tamaño poblacion/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generacion,
          para el cruce (probabilidad de seleccion proporcional
          a la funcion de evaluacion del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la funcion de evaluacion de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generacion.
        END
      END
    IF la poblacion ha convergido THEN
      Terminado := TRUE
    END
  END
```