

# COMPUTACIÓN BIOLÓGICA

Pedro Isasi<sup>1</sup>

<sup>1</sup>Departamento de Informática  
Universidad Carlos III de Madrid  
Avda. de la Universidad, 30. 28911 Leganés (Madrid). Spain  
email: isasi@ia.uc3m.es

Presentación

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- **Introducción a los Algoritmos Genéticos**
- Algoritmos Genéticos Canónicos
- Ejemplo
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- **Algoritmos Genéticos Canónicos**
- Ejemplo
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Genéticos Canónicos
- **Ejemplo**
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN

## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits	2 siguientes	2 siguientes	el último	
000	1			
001	2			
010	3			
011	4			
100	5			
101	Rnd[1, 5]			
110	Rnd[1, 5]			
111	Rnd[1, 5]			
	00	1		
	01	2		
	10	3		
	11	4		
		00	1	
		01	2	
		10	3	
		11	Rnd[1, 3]	
			0	1
			1	2

## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits	2 siguientes	2 siguientes	el último	
000	1			
001	2			
010	3			
011	4			
100	5			
101	Rnd[1, 5]			
110	Rnd[1, 5]			
111	Rnd[1, 5]			
	00	1		
	01	2		
	10	3		
	11	4		
		00	1	
		01	2	
		10	3	
		11	Rnd[1, 3]	
			0	1
			1	2

## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits	2 siguientes	2 siguientes	el último	
000	1			
001	2			
010	3			
011	4			
100	5			
101	Rnd[1, 5]			
110	Rnd[1, 5]			
111	Rnd[1, 5]			
	00	1		
	01	2		
	10	3		
	11	4		
		00	1	
		01	2	
		10	3	
		11	Rnd[1, 3]	
			0	1
			1	2

## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits	2 siguientes	2 siguientes	el último	
000	1			
001	2			
010	3			
011	4			
100	5			
101	Rnd[1, 5]			
110	Rnd[1, 5]			
111	Rnd[1, 5]			
	00	1		
	01	2		
	10	3		
	11	4		
		00	1	
		01	2	
		10	3	
		11	Rnd[1, 3]	
			0	1
			1	2



## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits		2 siguientes		2 siguientes		el último	
000	1	00	1	00	1	0	1
001	2	01	2	01	2	1	2
010	3	10	3	10	3		
011	4	11	4	11	Rnd[1, 3]		
100	5						
101	Rnd[1, 5]						
110	Rnd[1, 5]						
111	Rnd[1, 5]						

## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits	2 siguientes	2 siguientes	el último	
000	1			
001	2			
010	3			
011	4			
100	5			
101	Rnd[1, 5]			
110	Rnd[1, 5]			
111	Rnd[1, 5]			
	00	1		
	01	2		
	10	3		
	11	4		
		00	1	
		01	2	
		10	3	
		11	Rnd[1, 3]	
			0	1
			1	2

## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits	2 siguientes	2 siguientes	el último	
000	1			
001	2			
010	3			
011	4			
100	5			
101	Rnd[1, 5]			
110	Rnd[1, 5]			
111	Rnd[1, 5]			
	00	1		
	01	2		
	10	3		
	11	4		
		00	1	
		01	2	
		10	3	
		11	Rnd[1, 3]	
			0	1
			1	2

## CODIFICACIÓN PROBLEMA DEL VIAJANTE

- TSP 6 ciudades. Al principio se puede elegir entre 5, para codificarlo en binario se necesitan 3 bits:  $2^3 = 8 \geq 5$
- Luego hay que elegir entre 4, 2 bits:  $2^2 = 4 \geq 4$ , etc...
- El número total de bits es:  $3 + 2 + 2 + 1 = 8$
- El código genético será:

3 primeros bits	2 siguientes	2 siguientes	el último
000	1		
001	2		
010	3		
011	4		
100	5		
101	Rnd[1, 5]		
110	Rnd[1, 5]		
111	Rnd[1, 5]		

00	1
01	2
10	3
11	4

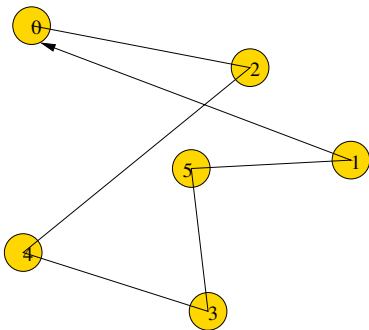
  

00	1
01	2
10	3
11	Rnd[1, 3]

0	1
1	2

# DESCODIFICACIÓN DE UN INDIVIDUO



Cromosoma= 00110010

La función de fitness es la distancia del recorrido, hay que minimizar

## EL PROBLEMA DE LA CODIFICACIÓN

- Para  $n = 6$  ciudades hay  $5! = 120$  soluciones posibles
- Con la codificación anterior hay  $2^8 = 256$  posibles individuos

### REDUNDANCIA

$\frac{120}{256}$  o sea el 53,13 %

- Si se aumenta la dimensión  $n = 20$ :
  - Soluciones:  $19! = 1,22 \times 10^{17}$
  - El tamaño del cromosoma necesario será:

$$\sum_{i=2}^{n-1} E(\log_2 i) = E(\log_2 19) + E(\log_2 18) + \dots + E(\log_2 2) = 64$$

- Siendo  $E(x)$  el entero inmediatamente superior al número real  $x$ , es decir:

$$E(x) = n | n > x, n \in \mathcal{N}, \nexists n' \in \mathcal{N} | n' > n$$

- Individuos:  $2^{64} = 1,84 \times 10^{19}$
- Redundancia:  $\frac{1,22 \times 10^{17}}{1,84 \times 10^{19}} \Rightarrow 99,34 \%$
- El procedimiento estará buscando de forma **TOTALMENTE** aleatoria

- Si  $n$  es la dimensión del problema el número de soluciones es:  $(n - 1)!$ , en binario se necesitan:

$$m = E(\log_2((n - 1)!))$$

- Lo que para un número de ciudades de por ejemplo 20 daría:

$$m = E(\log_2(19!)) = E(\log_2(1,22 \times 10^{17})) = 57$$

- Se ha pasado de 64 a 57 bits y la redundancia ha bajado de 99,34 % a 15,28 %

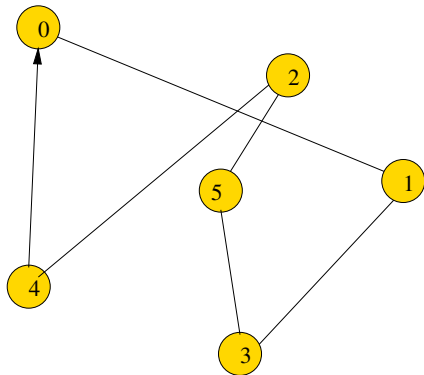


- 1 Convertir el cromosoma en un número decimal  $v$
- 2 Asignar a  $c$  el número de ciudades menos 1:  $c = n - 1$
- 3 Hallar el resto de la división entre  $v$  y  $c$  y asignárselo al vector con la solución:  $sol[c] = v \% c$
- 4 Asignarle a  $v$  el resultado entero de la división entre  $v$  y  $c$ :  $v = v / c$
- 5 Restarle 1 a  $c$ :  $c = c - 1$
- 6 Si  $c$  es mayor que 1 volver al paso 3

# CODIFICACIÓN BINARIA PARA PROBLEMA VIAJANTE

Para 6 ciudades:  $5! = 120$ ;  $E(\log_2 120) = 7$

Sea el individuo: 0101101:



Cromosoma= 0101101

Valor=45

$$\begin{array}{r} 45 \overline{) 5} \\ 0 \ 9 \overline{) 4} \\ 1 \ 2 \ \overline{) 3} \\ 2 \ 0 \ \overline{) 2} \\ 0 \ 0 \end{array}$$

Ciudades:: 0 1 3 5 2 4

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Genéticos Canónicos
- Ejemplo
- **Propiedades de los Algoritmos Genéticos**
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Genéticos Canónicos
- Ejemplo
- Propiedades de los Algoritmos Genéticos
- **Métodos de compartición y soluciones múltiples**
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Genéticos Canónicos
- Ejemplo
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- **Manejo de restricciones en Algoritmos Genéticos**
- Fundamentos de los Algoritmos Genéticos

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Genéticos Canónicos
- Ejemplo
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- **Fundamentos de los Algoritmos Genéticos**

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN

## 1 INTRODUCCIÓN

## 2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Genéticos Canónicos
- Ejemplo
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

## 3 COMPUTACIÓN EVOLUTIVA

## 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

## 5 BIOLOGÍA Y COMPUTACIÓN