



CIMPA-UCR

Clasificación Numérica usando Metaheurísticas de Optimización Combinatoria

Particionamiento Numérico usando Metaheurísticas de Optimización

1 Métodos de vecindarios

- 1.1 Sobrecalentamiento simulado
- 1.2 Búsqueda tabú

Optimizacion Combinatoria

$$F: S \rightarrow \mathbb{R}$$

S : espacio de estados (soluciones posibles)

Minimización: hallar i_{opt} tal que

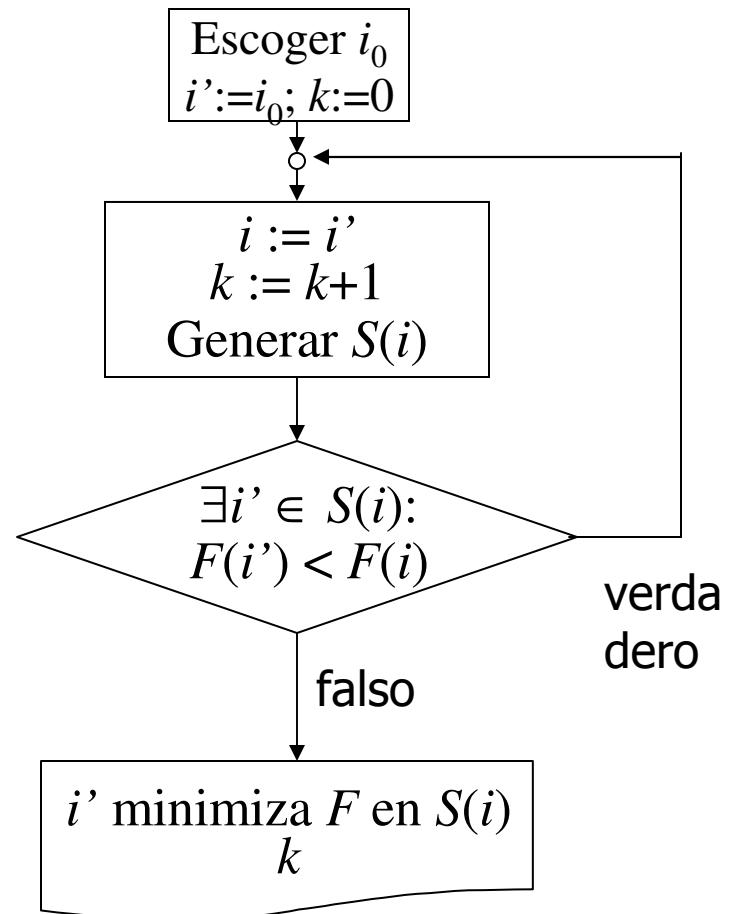
$$F(i_{\text{opt}}) \leq F(i), \text{ para todo } i \in S$$

Maximización: hallar i_{opt} tal que

$$F(i_{\text{opt}}) \geq F(i), \text{ para todo } i \in S$$

Búsqueda local

Defina una estructura de vecindarios: cómo generar un nuevo estado i' a partir del estado actual i

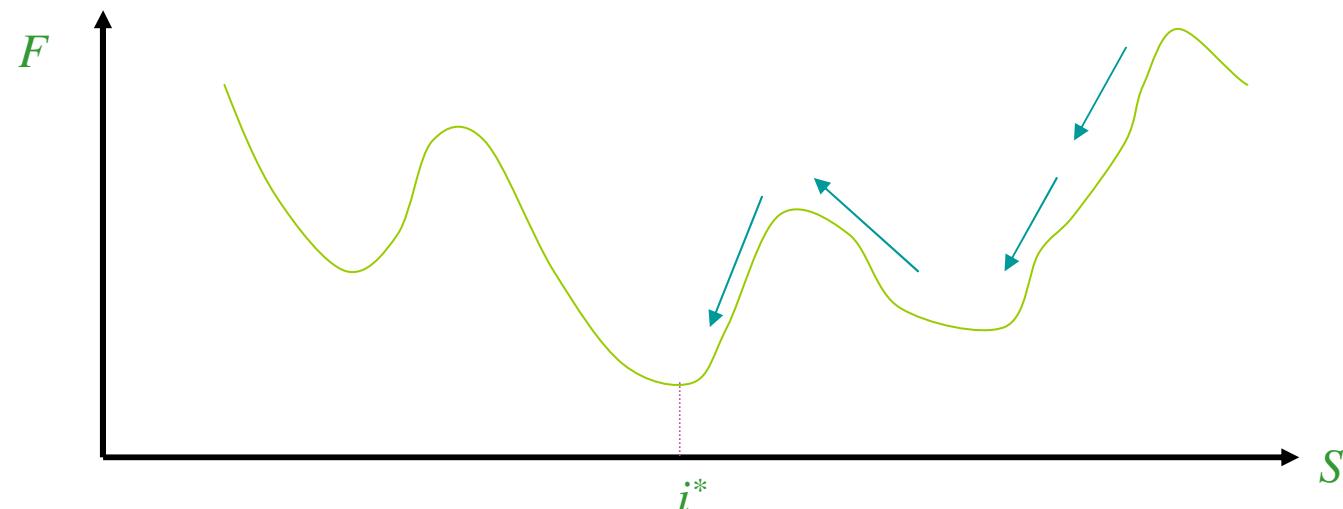


Sobrecalentamiento simulado

- También conocido como *Recocido simulado*, *Recocción simulada*, o *Temple simulado*. Son traducciones de *Simulated annealing*
- Método iterativo que usa un parámetro externo de **temperatura** T , que controla la aceptación de nuevos estados con un costo peor

Ilustración del SS

Habilidad para evitar mínimos locales



Regla de Metropolis

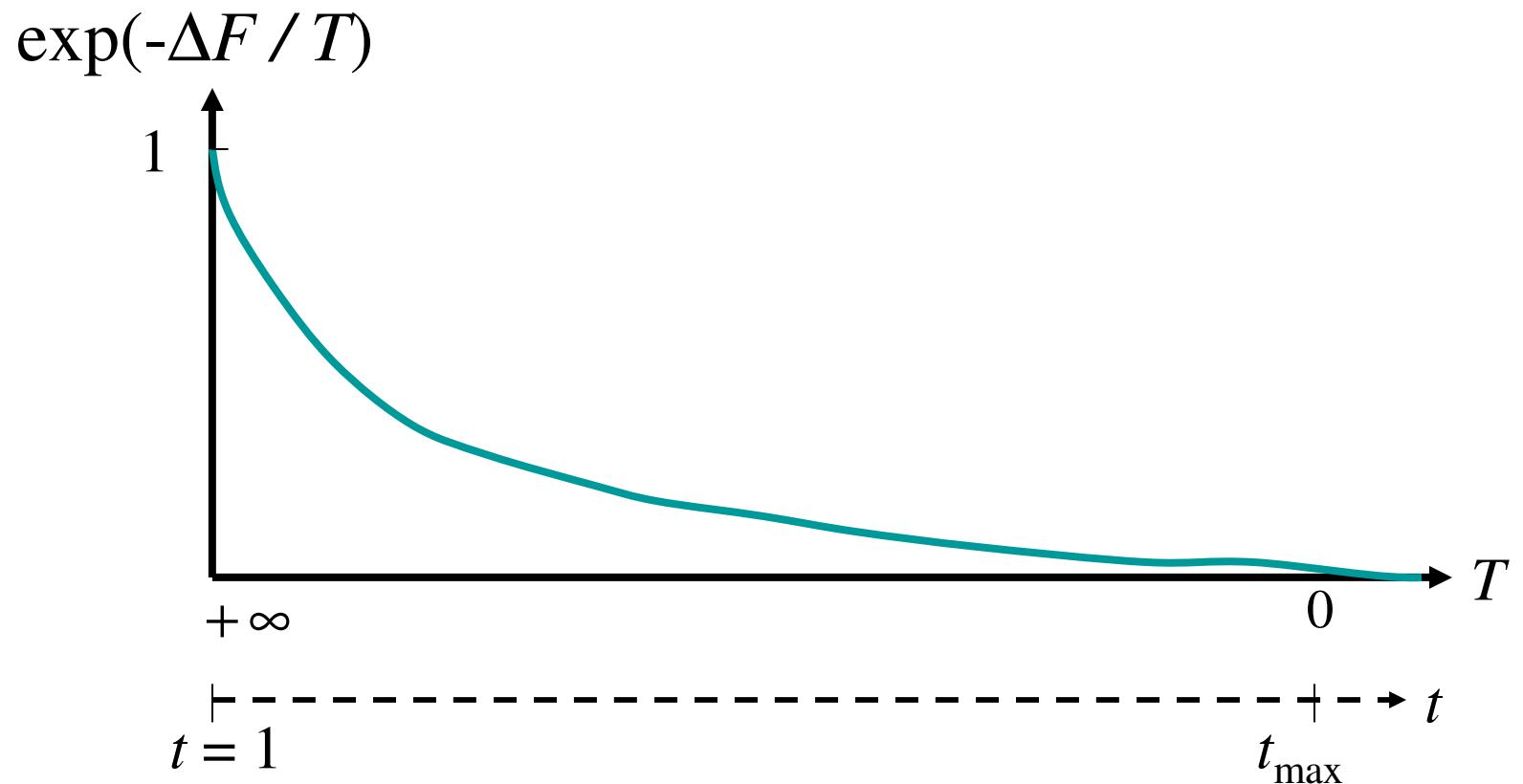
Un nuevo estado i' (en el vecindario del estado actual i) es aceptado si:

- El costo F decrece,
- Si no, puede ser aceptado con probabilidad

$$\exp(-\Delta F / T)$$

donde $\Delta F = F(i') - F(i)$

Ilustración de la R. Metrop.



Convergencia

- SS converge asintóticamente (con probabilidad 1) al mínimo global
- Una cadena de Markov (homogénea) es asociada a cada valor de T
- Condiciones:
reversibilidad y conexidad

Reversibilidad

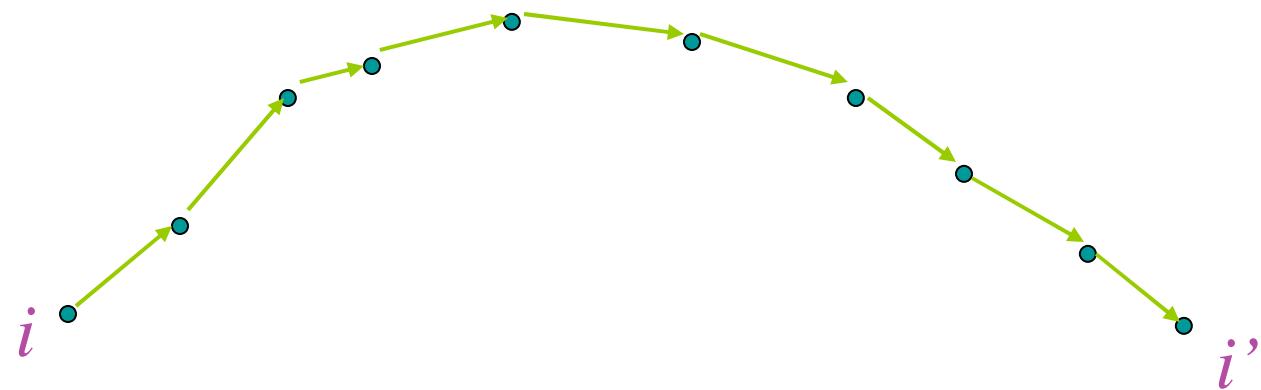
La probabilidad de regresar es la misma:

$$i \cdot \xrightarrow{G(i,i') = G_{ii'}} i'$$

$$i \cdot \xleftarrow{G(i',i) = G_{i'i}} i'$$

Conexidad

Hay una cadena finita entre dos estados,
con probabilidades de paso > 0



Implementación

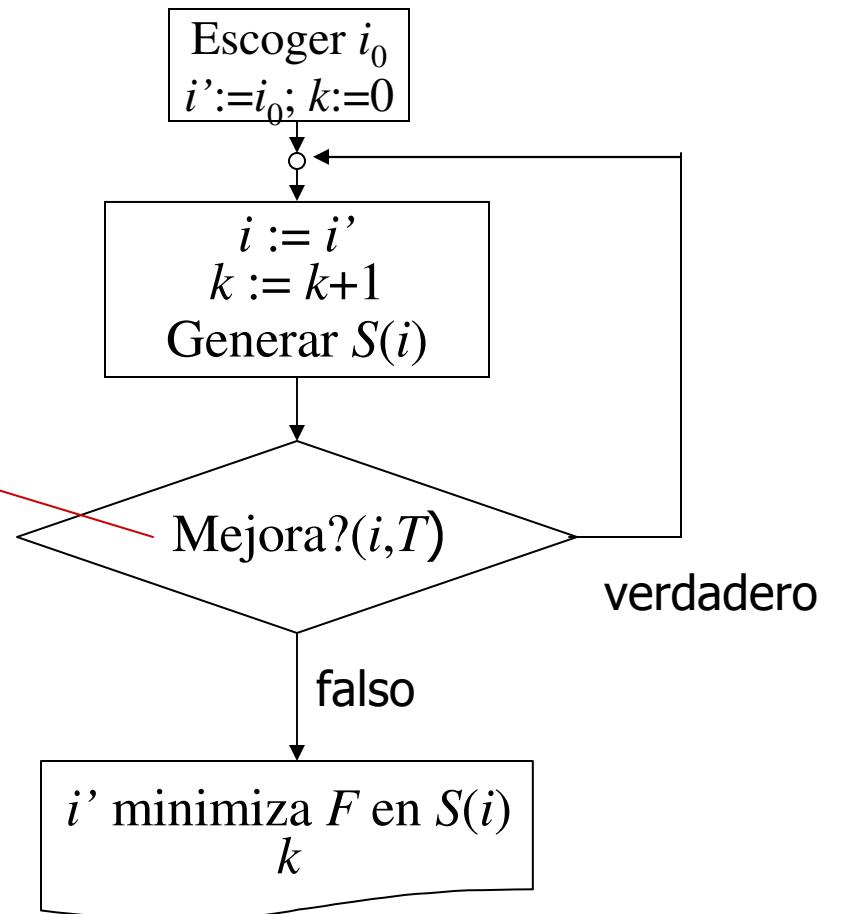
- T_0 : temperatura inicial (estimación mediante unas corridas en blanco con tasa inicial χ_0)
- T_f : temperatura (cuando $T \rightarrow 0$)
- $T_{t+1} := \gamma T_t$; $\gamma \in [0.8, 0.99]$ (enfriamiento)
- L_t : longitud de las cadenas de Markov
- ΔF : sencillo de calcular
- Reversibilidad y conexidad

Implementación

- T_0 : temperatura inicial (estimación mediante unas corridas en blanco con tasa inicial χ_0)
- T_f : temperatura (cuando $T \rightarrow 0$)
- $T_{t+1} := \gamma T_t$; $\gamma \in [0.8, 0.99]$ (enfriamiento)
- L_t : longitud de las cadenas de Markov
- ΔF : sencillo de calcular
- Reversibilidad y conexidad

Diagrama del SS

Mejora?(i, T):
 Aplicar la regla de Metropolis
 Encontrar un vecino y se acepta si es mejor, si no se acepta aleatoriamente de acuerdo con la temperatura actual T



SS en Particionamiento

A partir de una partición P generar P' como sigue:

1. Escoger al azar (unif. en $[1,n]$) un objeto $\mathbf{x} \in \Omega$
2. Escoger al azar(unif. en $[1,k]$) un índice de clase l
3. Poner a en la clase C_l

Nota: corresponde a lo que S. Régnier
llamaba **transferencia**

Características

- **Reversibilidad:** la probabilidad de $P \rightarrow P'$ es la misma que la probabilidad de $P' \rightarrow P$
- **Conexidad:** siempre es posible generar cualquier partición P' a partir de cualquier P (hay un # finito de transferencias)
- Vecindarios tienen el mismo tamaño: $n(k-1)$
- $G_{ss'} = 1/n(k-1)$

Simplificaciones

$$\Delta W = W(P) - W(P')$$
$$= \frac{|C_j|}{n(|C_j|-1)} \|\mathbf{g}_j - \mathbf{x}\|^2 - \frac{|C_l|}{n(|C_l|+1)} \|\mathbf{g}_l - \mathbf{x}\|^2$$

$$\mathbf{g}(C_j - \{\mathbf{x}\}) = \frac{1}{|C_j|-1} (|C_j| \mathbf{g}_j - \mathbf{x})$$

$$\mathbf{g}(C_l \cup \{\mathbf{x}\}) = \frac{1}{|C_l|+1} (|C_l| \mathbf{g}_l + \mathbf{x})$$

Búsqueda tabú

- Examen de vecindarios
- Maneja una **lista tabú**, que prohíbe regresar a estados previamente visitados
- La lista tabú contiene *movimientos*
- Se debe escoger el *mejor vecino*, que no sea tabú
- *Criterio de aspiración*: cómo eliminar el estado tabú a un buen vecino

Busqueda Tabú

- Heurística basada en el examen de un **vecindario** de un estado y la escogencia del **mejor vecino**
- Se maneja una **lista tabú** que prohíbe el acceso a estados similares a los estados recientemente visitados, procurando evitar los ciclos
- Hay mucha variantes que pueden mejorar los resultados (criterios de aspiración, estados elite, diversificación, intensificación, uso de muestras en los vecindarios, listas tabú de talla dinámica, ...)

Particionamiento con BT

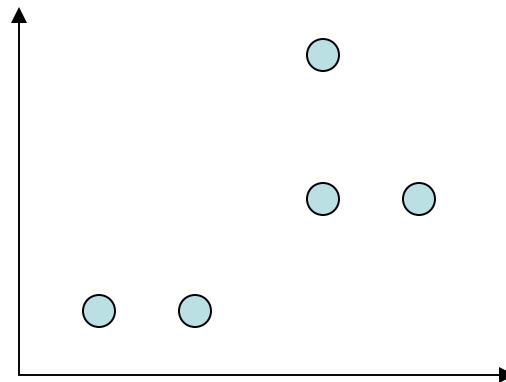
- Estado: partición P en k clases de Ω
- Criterio: minimizar W
- Movimiento: crear P' por transferencia de un único objeto a una nueva clase
- Lista tabú: indicador de la clase que contenía al objeto transferido

Particionamiento con BT

- Estado: partición P en k clases de Ω
- Criterio: minimizar W
- Movimiento: construir P' por la transferencia de un individuo hacia una nueva clase
- Lista tabú: indicadoras de la clase donde estaba el individuo transferido

Ilustración de la BT

- Sean los datos:



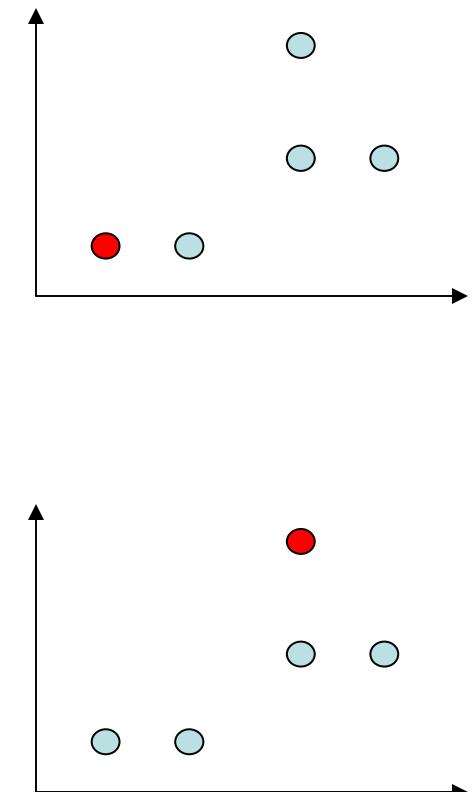
- Se quieren clasificar en 3 clases
- Partición inicial: (1 1 1 1 1) con $W = 3.52$



CIMPA-UCR

Iteración 1

Partición	W	
(21111) (31111)	2.15	
(12111) (13111)	3.00	
(11211) (11311)	3.35	
(11121) (11131)	2.55	
(11112) (11113)	2.15	



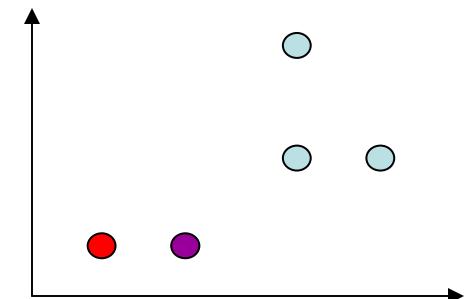
Lista tabú: 11111 10000



CIMPA-UCR

Iteración 2

Partición	W	
(11111)	3.52	Tabú
(21111)	2.15	Tabú
(32111) (31112)	1.07	
(33111)	1.17	
(31211)	2.13	
(31311)	3.13	
(31121)	1.47	
(31131)	3.07	
(31113)	2.87	



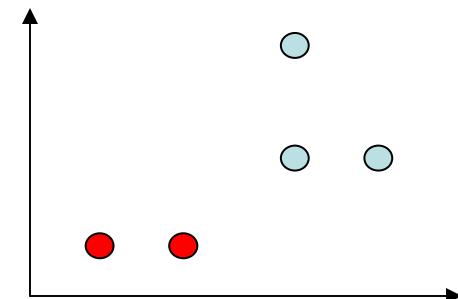
Lista tabú:
11111
10000
01000



CIMPA-UCR

Iteración 3

Partición	W	
(12111)	3.00	
(22111) (33111)	1.17	
(31111)	2.15	Tabú
(32211) (32112)	1.50	
(32311) (32131)	2.00	
(32121)	1.30	
(32113)	2.00	Tabú



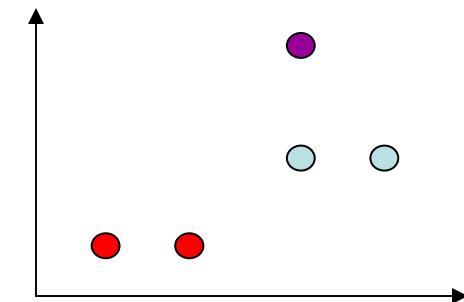
Lista tabú:
10000
01000
11000



CIMPA-UCR

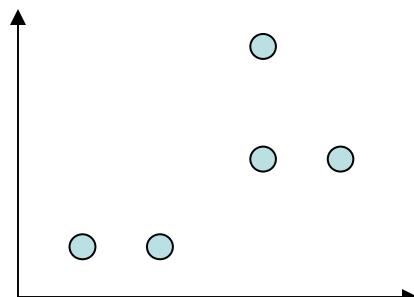
Iteración 4

Partición	W	
(13111)	3.00	
(23111) (32111)	1.07	Tabú
(31111)	2.15	Tabú
(33211)	1.10	
(33311)	2.07	
(33121)	0.50	
(33131)	2.13	
(33112)	0.30	
(33113)	2.33	

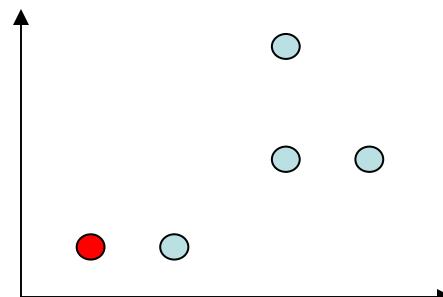


Lista tabú:
01000
11000
00001

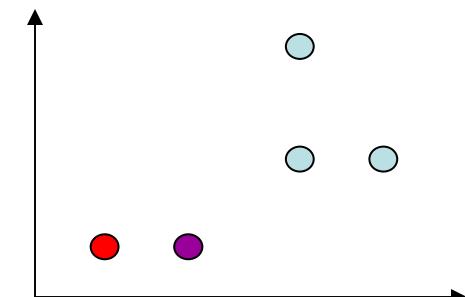
Evolución de Valores



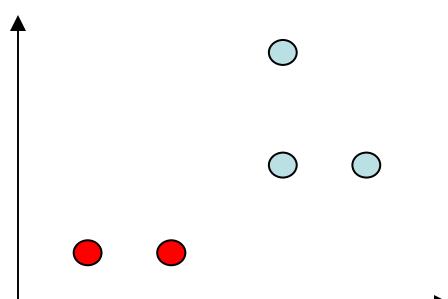
$$W = 3.52$$



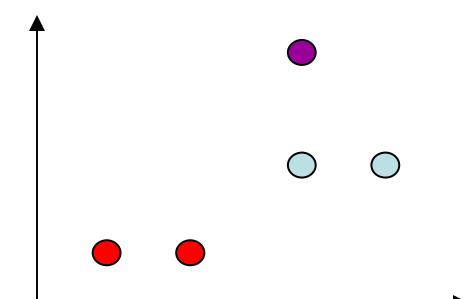
$$W = 2.15$$



$$W = 1.07$$



$$W = 1.17$$



$$W = 0.30$$

Más sobre BT

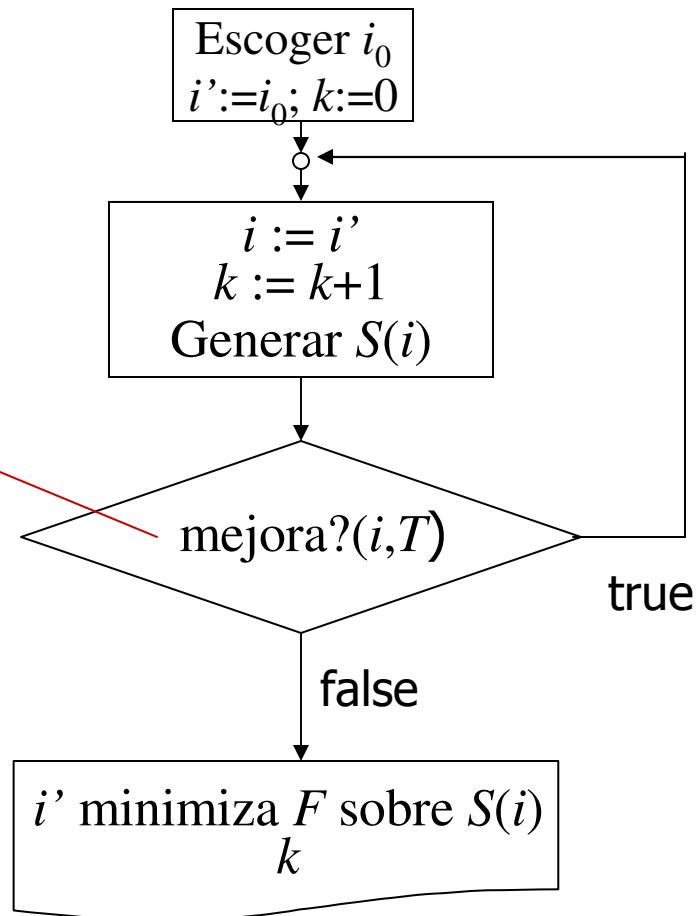
- *Estados elite*: registro de los mejores estados visitados
- *Intensificación*: en un sector “bueno”, continuar la búsqueda
- *Diversificación*: cambiar el sector del espacio de búsqueda
- Generación aleatoria de una *muestra* de vecinos

Diagrama de BT

Mejora?(i, T):

Depende de la lista tabú y del estado actual i

Analyza todo el vecindario o una muestra de él



Particionamiento Numérico

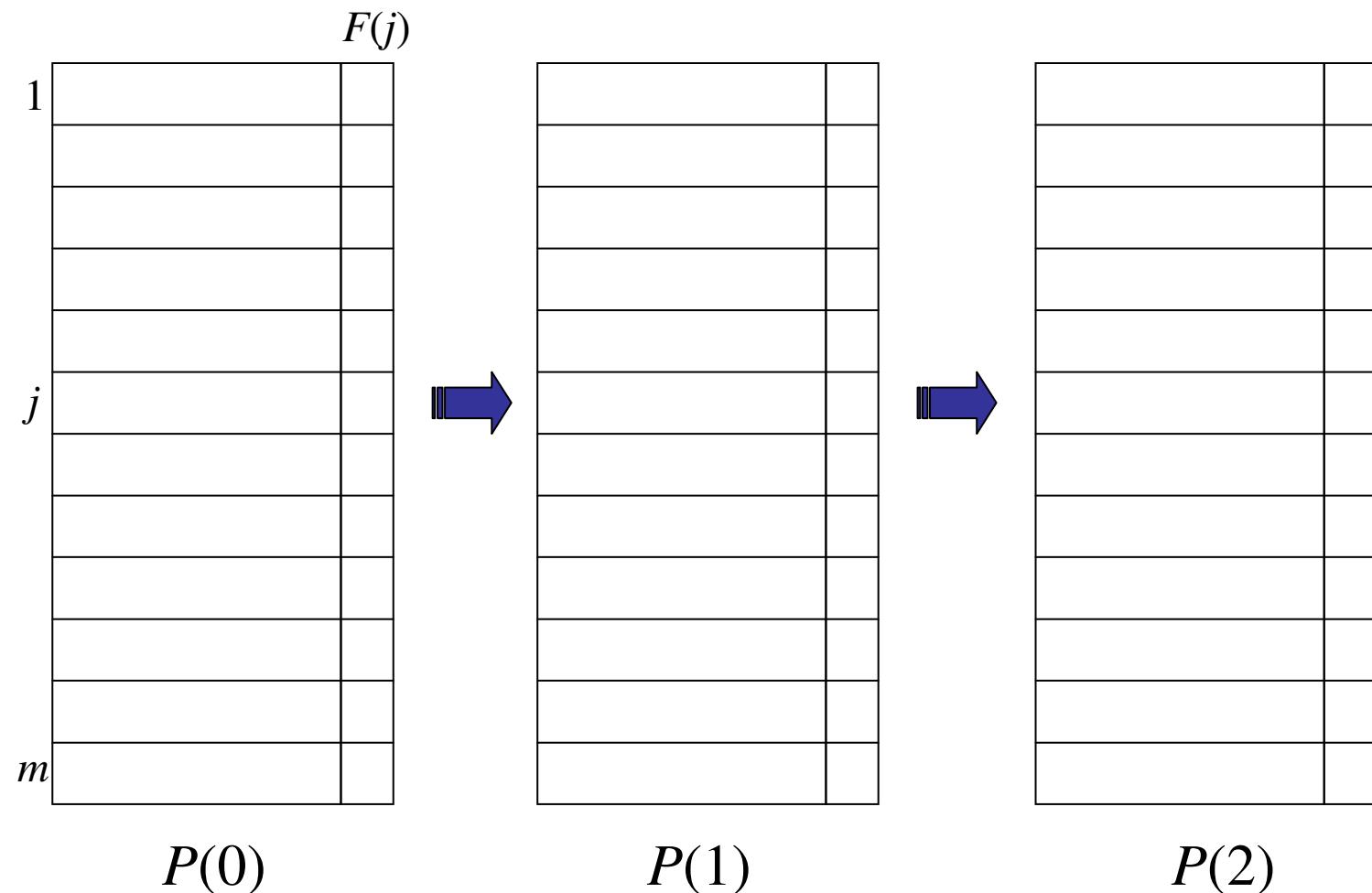
2 Métodos multiagentes

- 2.1 Algoritmos genéticos
- 2.2 Colonias de hormigas
- 2.3 Enjambres de partículas

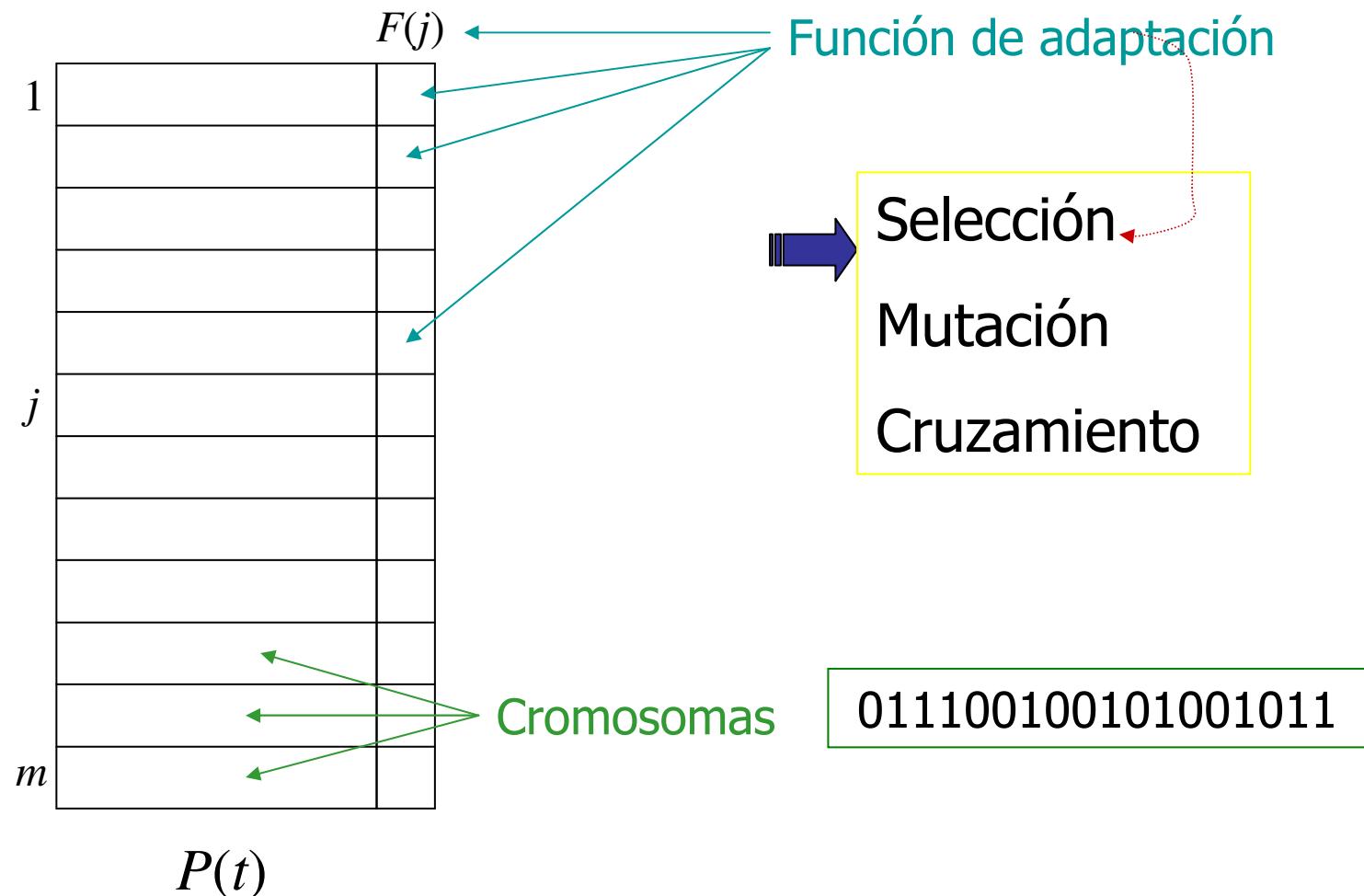
Algoritmos genéticos

- Algoritmos iterativos que simulan la evolución de las especies en la Naturaleza
- Principios:
 - Representación cromosómica de los estados Función de adaptación (al medio ambiente), Selección de los elementos mejor adaptados
 - Combinación de las características de los elementos usando operadores genéticos
 - Mutaciones
 - Cruzamientos

Evolución del AG

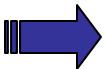


Elementos del AG



Operadores genéticos

Mutación:

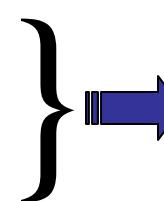
011100100101001011  011100100001001011

con probabilidad p_m

Cruzamiento:

011100100101001011

100100110101110111



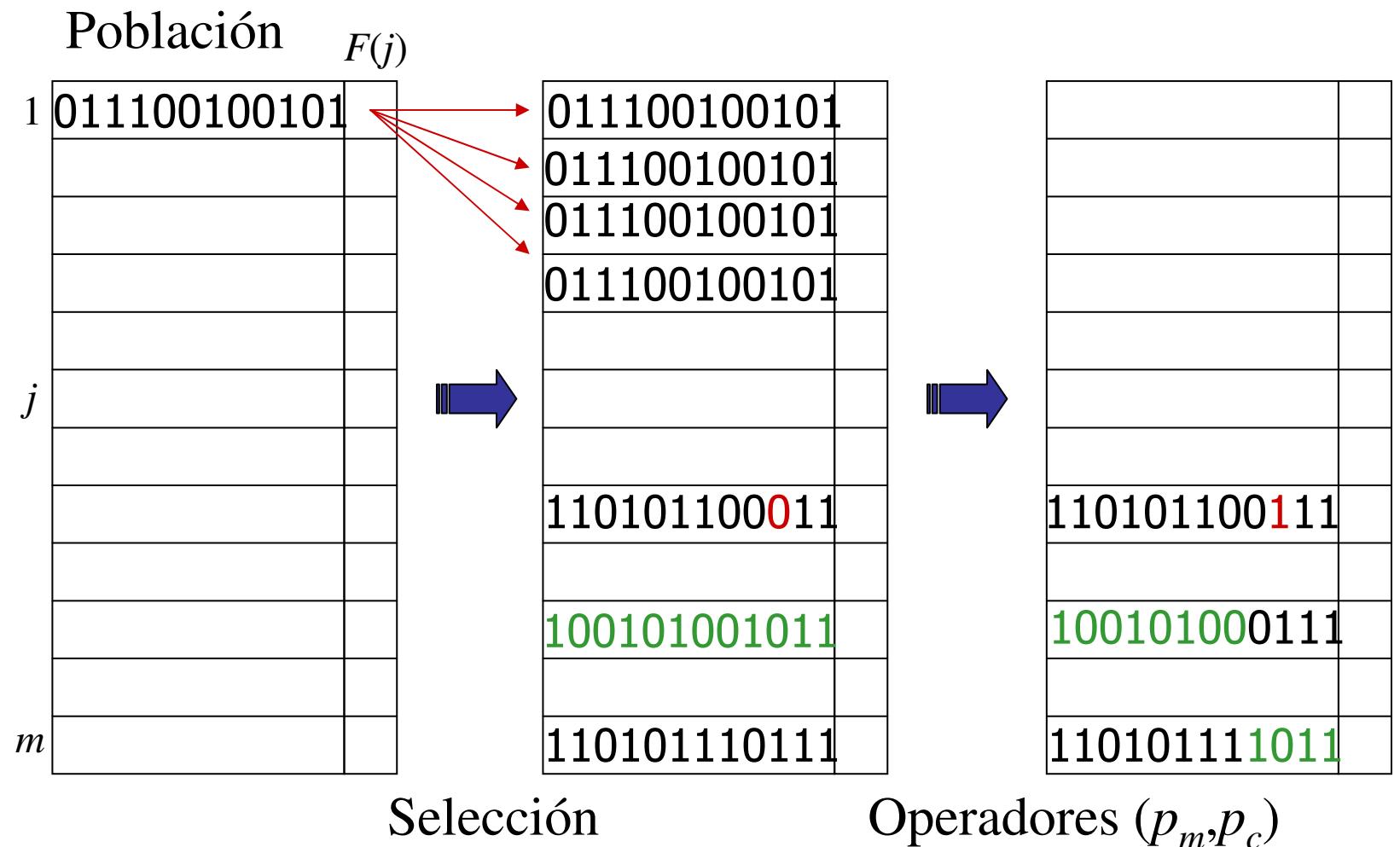
011100100101000111

100100110101111011

con probabilidad p_c



Evolución del AG



AG para particionamiento

- Estados: índices de clases

$$(2 \ 2 \ 3 \ 1 \ 1 \ 1 \ 3 \ 2)$$
$$\begin{matrix} \uparrow & \uparrow & \uparrow & & & & & \uparrow \\ x_1 & x_2 & x_3 & \dots & & & & x_n \end{matrix}$$

- Función de adaptación: B
- Selección: ruleta aleatoria proporcional a B
- Cruzamiento: con probabilidad p_c (cromos.)
- Mutación: con probabilidad p_m (alelo)

Cruzamiento

- Dados dos padres, escoger al azar una clase en el primero y copiarla en el segundo, para generar un hijo

$$\text{Ejem. : } P_1 = (112311232233)$$

$$P_2 = (112123122333)$$

$$P_1 : \{x_1, x_2, x_5, x_6\}, \{x_3, x_7, x_9, x_{10}\}, \{x_4, x_8, x_{11}, x_{12}\}$$

$$P_2 : \{x_1, x_2, x_4, x_7\}, \{x_3, x_5, x_8, x_9\}, \{x_6, x_{10}, x_{11}, x_{12}\}$$

Cruzamiento (2)

- Escoger al azar (unif.) la clase 2 de P_1 :

$$P_1 = (11\textcolor{red}{2}311\textcolor{red}{2}3\textcolor{black}{2}233)$$

- Copiar la clase 2 de P_1 en P_2 :

$$P_2 = (11\textcolor{blue}{2}123\textcolor{blue}{1}22333)$$

$$H = (11\textcolor{red}{2}123\textcolor{red}{2}2\textcolor{red}{2}233)$$

- Crear la partición:

$$H : \{x_1, x_2, x_4\}, \{x_3, x_5, x_7, x_8, x_9, x_{10}\}, \{x_6, x_{11}, x_{12}\}$$

Mutación

- Modificar un índice de clase escogido al azar:
- Ejem. $P = (12\textcolor{red}{1}331322)$
Si se escoge x_3 al azar (unif. in $[1, n]$) entonces la clase de pertenencia de x_3 es transformada en la clase 2 (índice escogido al azar unif. en $[1, k]$):
 $H = (12\textcolor{red}{2}331322)$

Ejemplo: Notas escolares

	Mate.	Ciencias	Historia	Latín	Ed. Fis.
Jean	6	6	5	5.5	8
Alain	8	8	8	8	9
Anne	6	7	11	9.5	11
Monique	14.5	14.5	15.5	15	8
Didier	14	14	12	12.5	10
André	11	10	5.5	7	13
Pierre	5.5	7	14	11.5	10
Brigitte	13	12.5	8.5	9.5	12
Evelyne	9	9.5	12.5	12	18

Componentes Principales

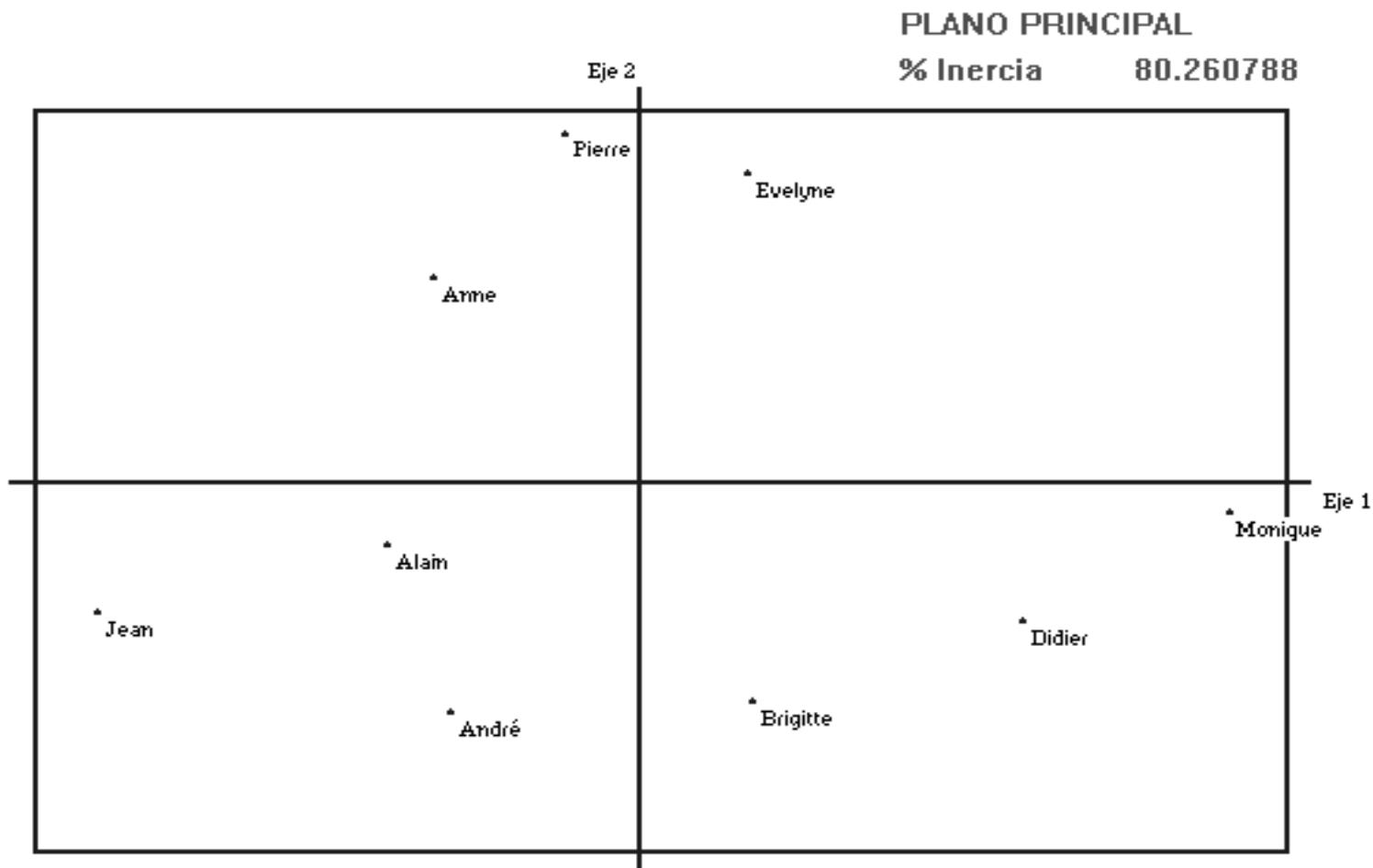


Ilustración del AG

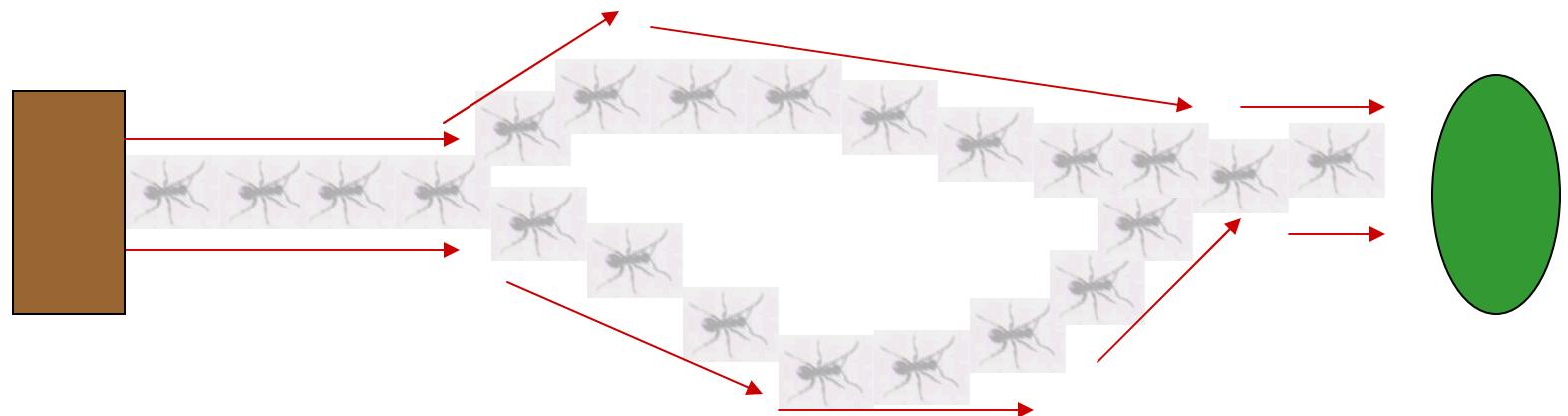
Iteración	3 clases			4 clases		
	Max	Min	Media	Max	Min	Media
0	24.62	0.97	12.23	34.66	3.96	16.88
20	30.64	19.60	28.90	38.50	14.94	33.58
40	30.64	15.14	28.05	37.66	27.01	36.52
60	30.64	11.03	27.48	38.50	26.34	37.31
80	32.16	14.50	28.69	38.50	27.13	35.80
100	32.16	23.51	30.19	38.50	14.41	35.62
120	32.16	13.09	29.73	38.50	14.41	35.59

Colonias de hormigas

- Metáfora: cómo hacen las hormigas para buscar alimento y evitar obstáculos
- *Reforzamiento*: trazo de feromonas
- Representación de las soluciones
- Se maneja una población de hormigas
- Cada hormiga construye o modifica una solución
- Heurística local o visibilidad
- Regla para actualizar las feromonas
- Regla probabilística de transición

Colonias de hormigas

- ACO: *Ant Colony Optimization*
- Metafora: manera como buscan alimento las hormigas
- Reforzamiento: trazo de feromona



ACO en Particionamiento

- M hormigas asociadas a M particiones
- Cada hormiga *modifica* su particion
- Heurística local: $\eta_{ij} = 1/d(i,j)$
- Trazo de feromonas:

$$\tau_{ij}(t)$$

ACO en Particionamiento

- Sea m una hormiga
- Sea i un objeto escogido al azar
- Se escoge un objeto j con probabilidad

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l=1}^n [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}$$

Trazo de feromonas

Para $i, j \in \Omega$:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho \sum_{m=1}^M \Delta^m \tau_{ij}(t+1)$$

con

$$\Delta^m \tau_{ij}(t+1) = \begin{cases} B(P^m)/I(\mathcal{N}) & \text{si } i, j \text{ están en la misma clase de } P^m \\ 0 & \text{sino.} \end{cases}$$

$\rho \in]0,1]$: parámetro de evaporación

Algoritmo ACOCLUS

Iniciar $\tau = \tau_0$; calcular η ; iniciar p

Dar P^1, \dots, P^m (al azar)

Para $t = 1$ hasta t_{\max} hacer

 Para $m = 1$ hasta M , hacer S veces

 Escoger al azar $i \in \Omega$

 Escoger j segun las reglas dadas

 Calcular $B(P^1), \dots, B(P^m)$ y actualizar τ

Enjambres de partículas

- *Particle Swarm Optimization* (PSO): Kennedy & Eberhardt (1995)
- <http://www.particleswarm.net>
- Modela el comportamiento social: cada individuo trata de mejorar de acuerdo con su propia experiencia y observando a sus vecinos
- Se maneja una población de M partículas o agentes en un espacio multidimensional

Principios de la PSO

Para cada partícula:

- Recordar mi mejor posición en el pasado
- Pregunto a mis vecinos por su mejor posición

Tendencias:

1. Seguir mi camino (*inercia*)
2. Regresar a mi mejor posición (*conservador*)
3. Imitar al líder (*imitación*)

PSO: modelo

$$v_m(t+1) = \alpha v(t) + \lambda_1 [z_m^*(t) - z_m(t)] + \lambda_2 [z^*(t) - z_m(t)]$$

$$z_m(t+1) = z_m(t) + v_m(t+1)$$

donde:

$$\lambda_1 = rand(0, \varphi_1), \lambda_2 = rand(0, \varphi_2)$$

$$\varphi = \varphi_1 + \varphi_2$$



CIMPA-UCR

PSO Modelo

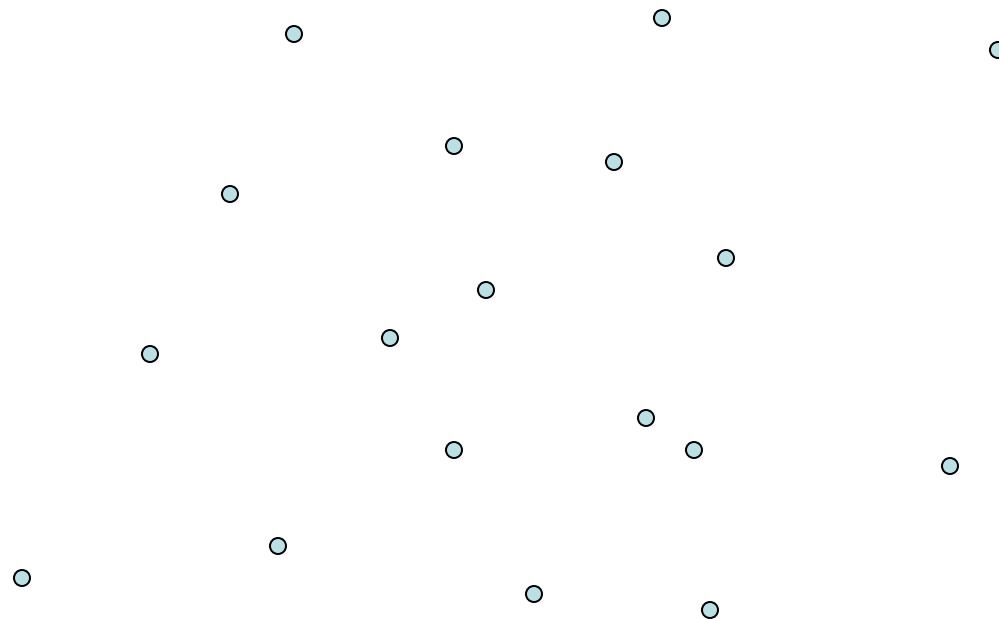
$$\begin{aligned} v_m(t+1) = & \alpha v_m(t) + && \text{Inercia} \\ & + rand(0, \varphi_1) [z_m^*(t) - z_m(t)] + && \text{Regresar} \\ & + rand(0, \varphi_2) [z^*(t) - z_m(t)] && \text{Imitación} \end{aligned}$$

$$z_m(t+1) = z_m(t) + v_m(t+1)$$



CIMPA-UCR

Ilustración de PSO





CIMPA-UCR

Ilustración de PSO

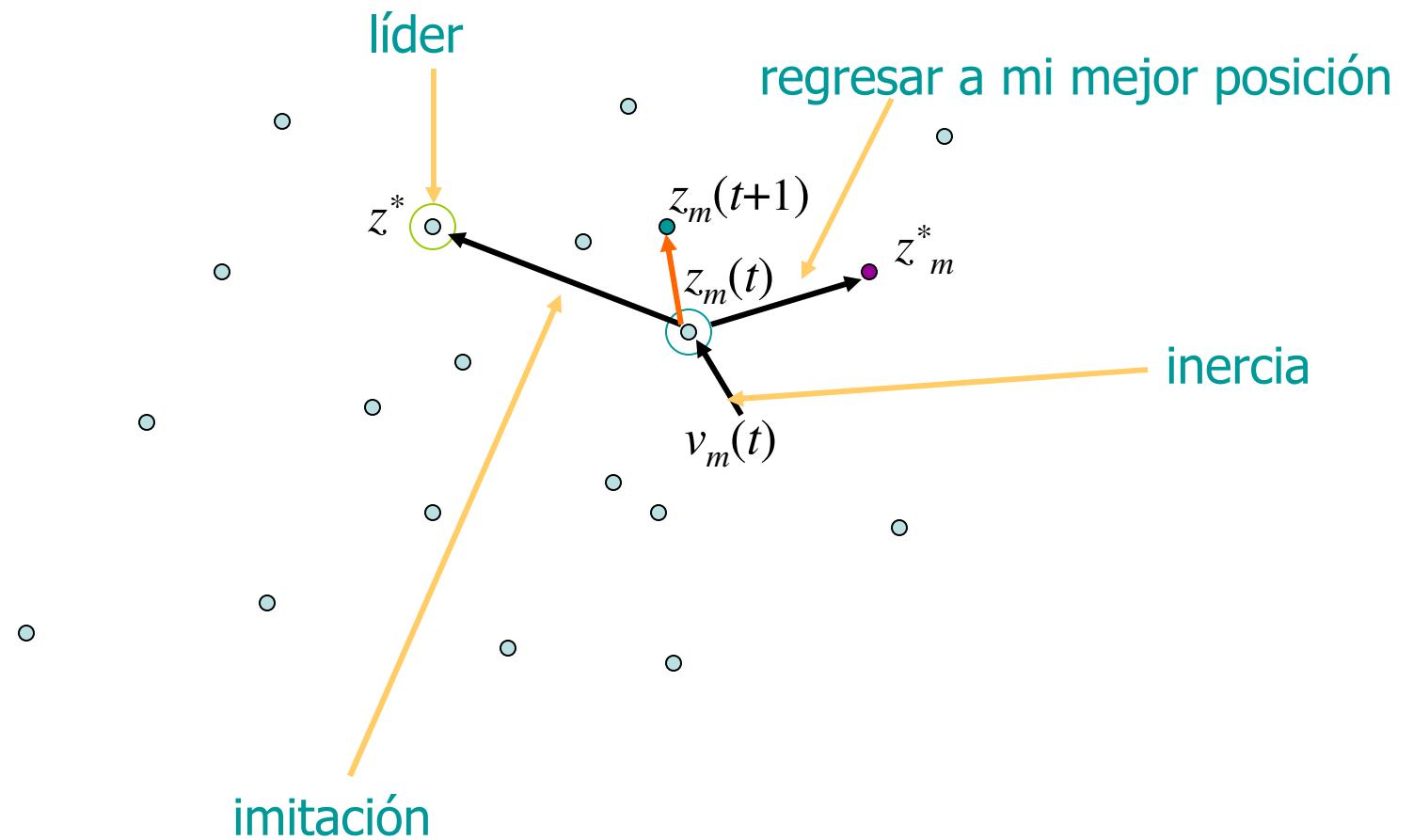
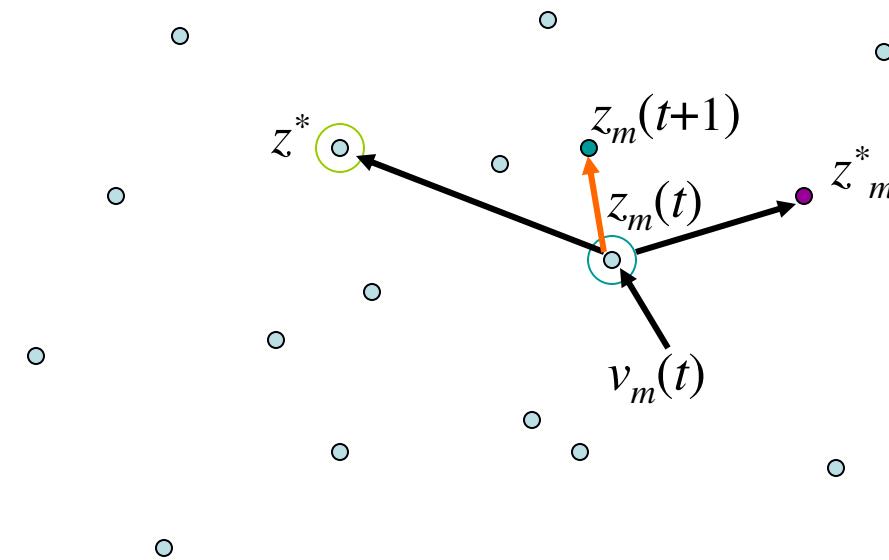


Ilustración de PSO



$$\begin{aligned}v_m(t+1) &= \alpha v_m(t) + \lambda_1 [z_m^*(t) - z_m(t)] + \lambda_2 [z^*(t) - z_m(t)] \\z_m(t+1) &= z_m(t) + v_m(t+1)\end{aligned}$$

Condiciones

Definir:

$$\kappa = [(e-2)(\varphi-1)-1] \left[1 + \frac{\sqrt{\varphi^2 - 4\varphi}}{\varphi-2} \right]$$

$$\chi = \frac{2\kappa}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}$$

Heuristica, resultados parciales: $\alpha = e - 2 \approx 0.718$

Condiciones para no divergencia:

$$\begin{cases} \kappa \in]0,1[\\ \alpha = \frac{1 + \chi(\varphi - 2)}{\varphi - 1} \end{cases}$$

Resultados empírico:

$$\kappa \geq 0.577, 4 < \varphi \leq 4.276$$

PSO en particionamiento

- Cada partícula es un conjunto de K centroides $\mathbf{g}_1, \dots, \mathbf{g}_K \in \mathbb{R}^p$
- Cada centroide tiene asociada una clase C_k por asignación de los objetos a la clase del centroide más cercano
- Los centroides se mueven en \mathbb{R}^p de acuerdo con los principios de la PSO y la partición se redefine por asignación
- Las partículas se mueven en \mathbb{R}^{Kp}

Algoritmo

Repeat for $t=1,2,\dots \text{ max_iter or convergencia}$

For $m=1:M$ (*particulas*)

For $k=1:K$ (*clases*)

use ecuación de PSO para cada
variable numérica

Asigne los objetos al centroide más cercano

Actualizar mejor posición de partícula m

Actualizar mejor partícula global

Parámetros

- $\alpha = 0.5$: coeficiente para $v_i(t)$
- $\varphi = \varphi_1 + \varphi_2 \geq 4$: suma de coefficients aleat.
- ε (tolerancia): para convergencia
- Máximo # de iteraciones (= 200)

Métodos comparados

- Nuestros métodos:
 - SA: Sobrecaleamiento simulado
 - TS: Búsqueda tabú
 - GA: Algoritmo genético
 - ACO: Colonias de hormigas
 - PSO: Enjambres de partículas
- Métodos clásicos:
 - KM: k -means
 - Ward: Clasif. Jerárquica Ascendente

Resultados comparativos

- Tablas de datos reales
- Tablas de datos simuladas

Resultados preliminares

Cuatro tablas de datos de la literatura:

- Notas escolares francesas (9 x 5)
- Peces de Amiard (23 x 16)
- Sociomatriz de Thomas (24 x 24)
- Iris de Fisher (150 x 4)

Parámetros de comparación

- Mínimo valor de inercia $W(P)$
- Porcentaje de veces alcanzado (tasa de atracción)



CIMPA-UCR

Resultados: Notas Escolares

Tabla 9 x 5

K	W	PSO #=100	SA #=150	TS #=1000	GA #=100	ACO #=25	KM #=10000	Ward
2	28.2	92	100	100	100	100	12	No
3	16.8	57	100	100	95	100	12	No
4	10.5	51	100	100	97	100	5	Sí
5	4.9	29	100	100	100	100	8	Sí



CIMPA-UCR

Resultados: Peces de Amiard

Tabla 23 x 16

K	W	PSO #=100	SA #=150	TS #=200	GA #=100	ACO #=25	KM #=10000	Ward
3	32213	51	100	100	87	100	8	No
4	18281	23	100	100	0	100	9	No
5	14497	6	100	97	0	68	1	Sí



CIMPA-UCR

Resultados: Sociomatriz de Thomas

Tabla 24 x 24

K	W	PSO $\#=100$	SA $\#=150$	TS $\#=200$	GA $\#=100$	ACO $\#=25$	KM $\#=10000$	Ward
3	271	7	100	100	85	100	2	No
4	235	7	100	100	24	96	0.15	No
5	202	7	100	98	0	84	0.02	No



CIMPA-UCR

Resultados: Iris de Fisher

Tabla 150 x 4

K	W	PSO $\#=100$	SA $\#=150$	TS $\#=1000$	GA $\#=100$	ACO $\#=25$	KM $\#=10000$	Ward
2	0.99	76	100	100	100	100	100	No
3	0.52	79	100	76	100	100	4	No
4	0.38	55	55	60	82	100	1	No
5	0.32	28	0	32	6	100	0.24	No

Experimentación

Simulación de Monte-Carlo:

Generar variables gaussianas $N(\mu, \sigma^2)$

- Número de objetos (100,500)
- Number of clusters (3,7)
- Varianza σ^2 (similares, diferentes)
- Cardinalidades de las clases (similares, diferentes)

Tablas de Datos: Experimento

- Factores

	Nivel 1	Nivel 2
# objetos	105	525
# clases	3	7
Card clases	$C_i = C_j$	$C_1 > C_j, j \neq 1$
Desv.est.	$s_i = s_j$	$s_2 > s_j, j \neq 2$

- Repeticiones : 100

Tablas de Datos

Name	Number of objects	Number of clusters	Clusters cardinalities	Standard deviation σ
AP0105cd03od06_e01	105	3	51,27,27	1,3,1
AP0105cd03oi06_e01	105	3	51,27,27	1,1,1
AP0105cd07od06_e01	105	7	51,9,9,9,9,9,9	1,3,1,1,1,1,1
AP0105cd07oi06_e01	105	7	51,9,9,9,9,9,9	1,1,1,1,1,1,1
AP0105ci03od06_e01	105	3	35,35,35	1,3,1
AP0105ci03oi06_e01	105	3	35,35,35	1,1,1
AP0105ci07od06_e01	105	7	15,15,15,15,15,15,15	1,3,1,1,1,1,1
AP0105ci07oi06_e01	105	7	15,15,15,15,15,15,15	1,1,1,1,1,1,1
AP0525cd03od06_e01	525	3	255,135,135	1,3,1
AP0525cd03oi06_e01	525	3	255,135,135	1,1,1
AP0525cd07od06_e01	525	7	255,45,45,45,45,45,45	1,3,1,1,1,1,1
AP0525cd07oi06_e01	525	7	255,45,45,45,45,45,45	1,1,1,1,1,1,1
AP0525ci03od06_e01	525	3	175,175,175	1,3,1
AP0525ci03oi06_e01	525	3	175,175,175	1,3,1
AP0525ci07od06_e01	525	7	75,75,75,75,75,75,75	1,3,1,1,1,1,1
AP0525ci07oi06_e01	525	7	75,75,75,75,75,75,75	1,1,1,1,1,1,1



CIMPA-UCR

Generación de las Tablas de Datos

Clasificación Numérica - Métodos Matemáticos de Optimización Combinatoria

*.def

```
6 : cantidad de variables
3 : cantidad de clases
1 51 1 : indice de la clase, cantidad de individuos y desviación estándar de las variables para esa clase
2 27 1 : idem
3 27 1 : idem
```

*.esp

```
6 : cantidad de variables
3 : cantidad de clases
51 5.02 1.00 9.54 1.00 0.47 1.00 5.03 1.00 2.92 1.00 4.29 1.00
27 1.99 1.00 4.39 1.00 2.38 1.00 8.88 1.00 2.47 1.00 8.62 1.00
27 6.59 1.00 1.60 1.00 9.09 1.00 1.78 1.00 4.02 1.00 4.82 1.00
9.10751554800000E-0002 : semilla que se utilizó para generar
                           μ de cada variable de cada clase
```

Las medias (μ) se obtuvieron generando un número aleatorio entre 0 y 1, y el valor obtenido fue multiplicado por 10.

*.dat

```
105 : cantidad de individuos
6 : cantidad de variables
4.60 0.74 0.96 4.59 2.74 4.38 : valores para cada una de las variables
5.50 0.58 1.61 6.03 3.45 5.22
4.28 6.85 1.61 5.67 0.74 6.06
.
.
.
3.91 3.36 9.08 2.17 3.01 3.83
5.64 0.54 10.14 3.89 5.73 4.02
5.00677923021060E+0000 :  $W_{orig} = W$  de la clasificación predefinida para los individuos
3.11728207200000E-0002 : semilla utilizada en el generador de números aleatorios
```



CIMPA-UCR

Parámetros

Notación: n es la cantidad de individuos
 k es la cantidad de clases

SA

Parámetro	Valor
Tasa de aceptación inicial χ_0	95%
Tasa de enfriamiento γ	90%
Largo de las cadenas de Markov L_c	$100 * n * (k - 1)$
Límite para detener las cadenas de Markov $nLimit$	$10 * n * (k - 1)$
Número de iteraciones del valor de la temperatura $nTemp$	150
Número máximo de temperaturas sin aplicar la regla de Metropolis $ncad$	4

TS

Parámetro	Valor
Longitud de la lista tabú $ T $	7
Cantidad máxima de iteraciones	$= n * k * T $

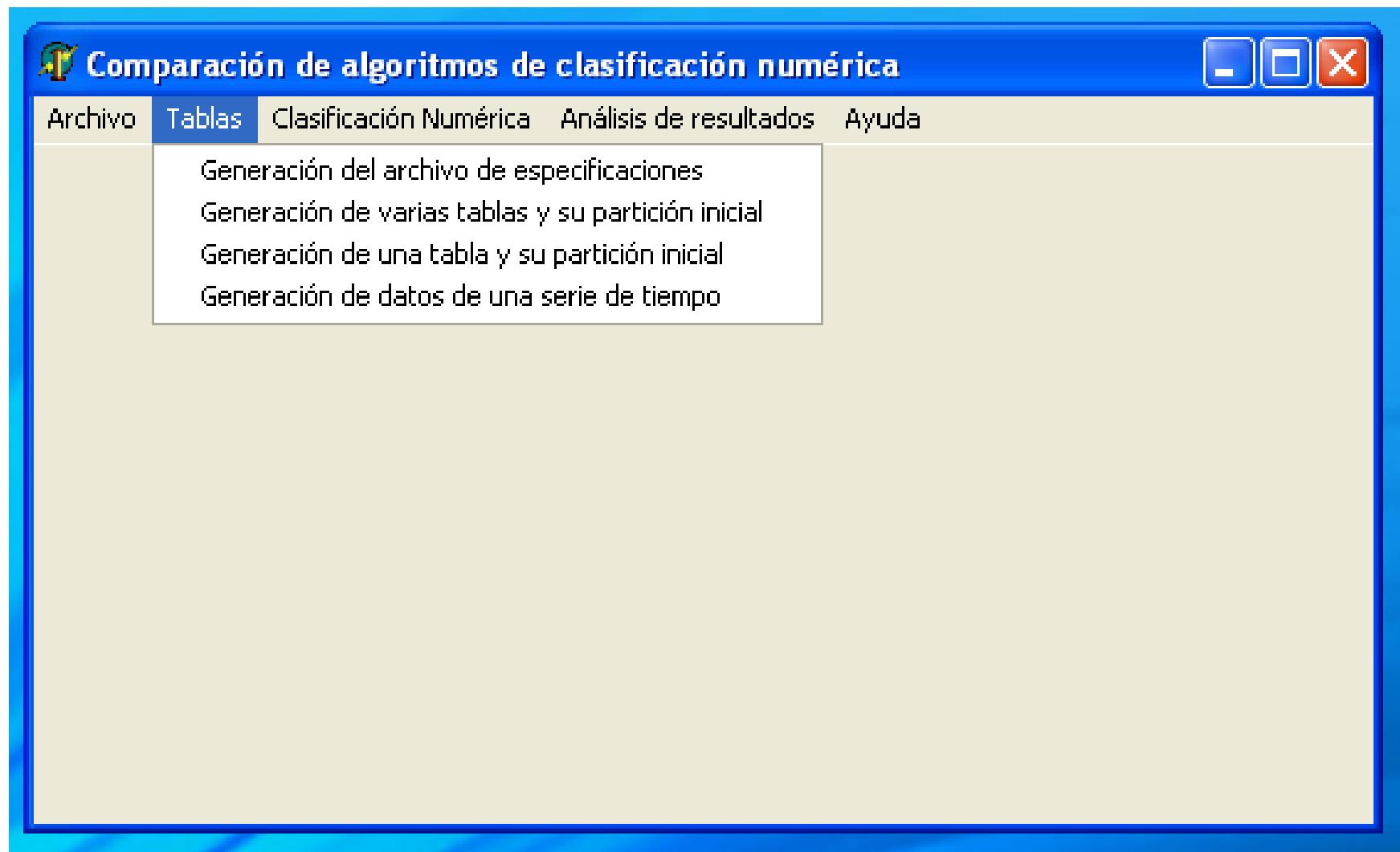
GA

Parámetro	Valor
Tamaño de la población $T_{population}$	40
Probabilidad de cruzamiento forzado p_c	0.25
Probabilidad de mutación p_m	0.001
Número de iteraciones	tamaño de la población * $k * \log(n)$
Frecuencia para aplicar nubes dinámicas	cada 10 iteraciones
Tipos de cruzamiento	cruzamiento forzado
Tipos de mutaciones	mutación simple

Desarrollo de Software

- Programación en Delphi®
- Implementación de nuestros métodos aplicando las Metaheurísticas de Optimización
- Generación aleatoria de tablas de datos
- Comparaciones: tablas resumen

Generación de Datos





CIMPA-UCR

Métodos

 Comparación de algoritmos de clasificación numérica

- □ ×

Archivo Tablas Clasificación Numérica Análisis de resultados Ayuda

- Búsqueda Tabú
- Génetico
- Sobrecalentamiento
- Nubes dinámicas
- Grasp
- Hormigas
- Enjambre de partículas
- Procesamiento en lotes



CIMPA-UCR

Ejemplo: SS

Clasificación Numérica usando Metaheurísticas de Optimización Combinatoria

Clasificación por particiones usando sobrecaleamiento simulado

Archivos

Filtro para archivo de datos: C:\Documents and Settings\Javier Trejos\Mis documentos\Articulos\

Inicio: Cantidad de réplicas: Partición inicial en archivo

Parámetros

Chi_0	<input type="text" value="0.9"/>	nlimit	<input type="text" value="500"/>	nCad	<input type="text" value="3"/>
Tasa de decrecimiento de la temperatura	<input type="text" value="0.9"/>	No. Clases	<input type="text" value="3"/>		
Largo de la cadenas	<input type="text" value="20000"/>	Número máximo de iteraciones: <input type="text" value="100"/>			

Seguimiento

Iteración	Varianza
-----------	----------

Ejemplo: AG & ACO



CIMPA-UCR

Clasificación Numérica usando Metaheurísticas de Optimización Combinatoria

Algoritmo genético de particionamiento - Nubes dinámicas en cada iterac...

Archivos
Filtro para archivo de datos C:\Documents and Settings\Javier Trejos\Mis documentos\Articulos\

Procesando el archivo IRIS.DAT

Parámetros
No. Clases 3 P. Cruzamiento 0.25
Tamaño población 45 P. Mutación 0.001
Máx. generaciones 2000

Operadores genéticos
 Cruzamiento simple
 Cruzamiento forzado
 Mutación simple
 Mutación débil

Algoritmo Colonia de Hormigas para particionamiento

Archivos
Filtro para archivo de datos

Inicio 1 Cantidad de réplicas 100 Partición inicial en archivo

Parámetros
No. Clases 4 Valor inicial del feronomo 0.001
No. Max. de Iteraciones 25 Potencia del trazo del feronomo 1
Tamaño de la población 20 Tasa de evaporación 0.5
Factor de transferencias 3 Potencia de la visibilidad 0.8
Tipo de distancia Euclidea clásica q (tau.proba) 0.95
Tolerancia 0.00001

Parámetros de comparación:

- Mínimo valor de inercia $W(P)$
- Porcentaje de veces alcanzado
- Valor promedio de W en las ejecuciones
- % de mala clasificación

Resultados 1

n	K	$ C_k $	σ
105	3	=	=

$W_0 = 5.542$	W^*	τ	W prom	% mal cl.
Sob.simul.	5.422	100%	5.422	0%
Bus.Tabu	5.422	99%	5.533	0.5%
Alg.Gen.	5.422	100%	5.422	0%
Hormigas	5.422	100%	5.422	0%
K-medias	5.422	91%	5.422	3.5%
Ward	5.42			0%

Resultados 2

n	K	$ C_k $	σ
105	7	=	=

$W_0 = 5.202$	W^*	τ	W prom	% mal cl.
Sob.simul.	5.146	100%	5.146	0.95%
Bus.Tabu	5.146	74%	5.968	5.86%
Alg.Gen.	5.146	82%	5.285	3.16%
Hormigas	5.146	100%	5.146	0.95%
K-medias	5.146	19%	7.777	14.99%
Ward	5.15			0.95%

Resultados 3

n	K	$ C_k $	σ
525	3	=	=

$W_0 = 5.993$	W^*	τ	W prom	% mal cl.
Sob.simul.	5.993	100%	5.993	0%
Bus.Tabu	5.993	100%	5.993	0%
Alg.Gen.	5.993	100%	5.993	0%
Hormigas	5.993	100%	5.993	0%
K-medias	5.993	98%	6.146	0.70%
Ward	5.99			0%

Resultados 4

n	K	$ C_k $	σ
525	7	=	=

$W_0 = 5.339$	W^*	τ	W prom	% mal cl.
Sob.simul.	5.339	100%	5.339	0%
Bus.Tabu	5.339	82%	5.872	3.73
Alg.Gen.	5.339	88%	5.654	2.15
Hormigas	5.339	100%	5.339	
K-medias	5.339	45%	7.195	10.74
Ward	5.34			0%

Resultados 5

n	K	$ C_k $	σ
105	3	=	\neq

$W_0 = 14.33$	W^*	τ	W prom	% mal cl.
Sob.simul.	13.15	100%	13.15	13.33%
Bus.Tabu	13.15	99%	13.18	13.68%
Alg.Gen.	13.15	100%	13.15	13.33%
Hormigas	13.15	100%	13.15	13.33%
K-medias	13.15	13%	13.50	20.47%
Ward	13.15			6.67%

Resultados 6

n	K	$ C_k $	σ
105	7	=	\neq

$W_0 = 11.305$	W^*	τ	W prom	% mal cl.
Sob.simul.	9.895	100%	9.895	14.29%
Bus.Tabu	9.895	51%	10.097	11.56%
Alg.Gen.	9.895	69%	9.947	13.21%
Hormigas	9.895	75%	9.897	14.29%
K-medias	9.895	1%	12.790	18.68%
Ward	10.17			14.29%

Resultados 7

n	K	$ C_k $	σ
525	3	=	\neq

$W_0 = 16.169$	W^*	τ	W prom	% mal cl.
Sob.simul.	15.809	100%	15.809	4.95%
Bus.Tabu	15.809	51%	15.810	4.58%
Alg.Gen.	15.809	82%	16.006	6.14%
Hormigas	15.809	99%	15.810	
K-medias	15.81	2%	16.139	7.35%
Ward	16.41			1.33%

Resultados 8

n	K	$ C_k $	σ
525	7	=	\neq

$W_0 = 9.37$	W^*	τ	W prom	% mal cl.
Sob.simul.	8.261	100%	8.261	8.95%
Bus.Tabu	8.261	100%	8.261	8.95%
Alg.Gen.	8.261	94%	8.356	9.59%
Hormigas	8.261	100%	8.261	8.95%
K-medias	8.261	53%	8.626	11.39%
Ward	9.37			14.67%

Resultados 9

n	K	$ C_k $	σ
105	3	\neq	$=$

$W_0 = 5.007$	W^*	τ	W prom	% mal cl.
Sob.simul.	5.007	100%	5.007	0%
Bus.Tabu	5.007	100%	5.007	0%
Alg.Gen.	5.007	100%	5.007	0%
Hormigas	5.007	100%	5.007	0%
K-medias	5.007	91%	6.375	3.65%
Ward	5.01			0%

Resultados 10

n	K	$ C_k $	σ
105	7	\neq	$=$

$W_0 = 5.545$	W^*	τ	W prom	% mal cl.
Sob.simul.	6.991	62%	7.769	34.12%
Bus.Tabu	6.991	1%	11.723	52.14%
Alg.Gen.	5.545	35%	6.772	17.50%
Hormigas	5.545	36%	6.745	14.67
K-medias	5.545	3%	10.263	25.67%
Ward	5.55			0%

Resultados 11

n	K	$ C_k $	σ
525	3	\neq	$=$

$W_0 = 5.672$	W^*	τ	W prom	% mal cl.
Sob.simul.	5.672	8%	5.701	0.25%
Bus.Tabu	5.672	100%	5.672	0%
Alg.Gen.	5.672	100%	5.672	0%
Hormigas	5.672	100%	5.672	0%
K-medias	5.672	96%	5.891	2.25%
Ward	5.67			25.71%

Resultados 12

n	K	$ C_k $	σ
525	7	\neq	$=$

$W_0 = 5.648$	W^*	τ	W prom	% mal cl.
Sob.simul.	8.105	100%	8.105	49.14%
Bus.Tabu	8.105	1%	10.714	58.23%
Alg.Gen.	5.648	22%	6.893	23.42%
Hormigas	5.648	38%	6.350	
K-medias	5.648	2%	8.482	32.43%
Ward	5.66			0.19%

Resultados 13

n	K	$ C_k $	σ
105	3	\neq	\neq

$W_0 = 12.769$	W^*	τ	W prom	% mal cl.
Sob.simul.	11.734	100%	11.734	2.86%
Bus.Tabu	11.734	100%	11.734	2.86%
Alg.Gen.	11.734	100%	11.734	2.86%
Hormigas	11.734	100%	11.734	2.86%
K-medias	11.734	95%	12.225	4.93%
Ward	11.86			3.81%

Resultados 14

n	K	$ C_k $	σ
105	7	\neq	\neq

$W_0 =$	W^*	τ	W prom	% mal cl.
Sob.simul.	8.654	100%	8.654	29.52%
Bus.Tabu	8.654	40%	9.819	37.27%
Alg.Gen.	7.625	37%	8.365	17.99%
Hormigas	7.625	85%	7.681	2.65%
K-medias	7.625	6%	9.758	25.87%
Ward	7.69			0.95%

Resultados 15

n	K	$ C_k $	σ
525	3	\neq	\neq

$W_0 = 14.196$	W^*	τ	W prom	% mal cl.
Sob.simul.	13.819	3%	13.819	2.13%
Bus.Tabu	13.819	100%	13.819	2.29%
Alg.Gen.	13.819	100%	13.819	2.29%
Hormigas	13.819	100%	13.819	2.25%
K-medias	13.819	59%	14.116	2.80%
Ward	14.2		14.200	0%

Resultados 16

n	K	$ C_k $	σ
525	7	\neq	\neq

$W_0 = 8.137$	W^*	τ	W prom	% mal cl.
Sob.simul.	8.497	10%	9.760	47.48%
Bus.Tabu	8.518	1%	10.165	50.89%
Alg.Gen.	7.456	21%	8.534	25.26%
Hormigas	7.456	54%	7.722	9.22%
K-medias	7.463	2%	9.269	28.91%
Ward	8.000			6.10%



CIMPA-UCR

Clasificación Numérica usando Metaheurísticas de Optimización Combinatoria

Frecuencia de W^*

n	K	W^*	PSO	SA	TS	GA	ACO	kM	Ward	
Equal cardinalities										
<i>Equal variances</i>										
105	3	5,42	100	100	99	100	100	91	yes	
105	7	5,15	1	100	74	82	100	19	yes	
525	3	5,99	94	100	100	100	100	98	yes	
525	7	5,34	1	100	82	88	100	45	yes	
<i>Different variances</i>										
105	3	13,15	1	100	99	100	100	13	no	
105	7	9,90	0	100	51	69	75	1	no	
525	3	15,81	1	100	51	82	99	2	no	
525	7	8,26	0	100	100	94	100	53	no	
Different cardinalities										
<i>Equal variances</i>										
105	3	5.01	99	100	100	100	100	91	yes	
105	7	5.55	1	0	0	35	36	3	yes	
525	3	5.67	84	8	100	100	100	95	yes	
525	7	5.65	1	0	0	22	38	2	yes	
<i>Different variances</i>										
105	3	11.73	12	100	100	100	100	95	no	
105	7	7.63	0	0	0	37	85	6	no	
525	3	13.82	1	3	100	100	100	59	no	
525	7	7.46	0	0	0	21	54	0	no	



CIMPA-UCR

Clasificación Numérica usando Metaheurísticas de Optimización Combinatoria

Valor medio de W

n	K	W^*	PSO	SA	TS	GA	ACO	kM	Ward
<i>Equal cardinalities</i>									
<i>Equal variances</i>									
105	3	5.42	5.42	5.42	5.53	5.42	5.42	6.42	5.42
105	7	5.15	6.21	5.15	5.97	5.29	5.15	7.78	5.15
525	3	5.99	5.99	5.99	5.99	5.99	5.99	6.15	5.99
525	7	5.34	6.87	5.34	5.87	5.65	5.34	7.20	5.34
<i>Different variances</i>									
105	3	13.15	13.31	13.15	13.18	13.15	13.15	13.50	13.85
105	7	9.90	11.26	9.90	10.10	9.95	9.90	12.79	10.17
525	3	15.81	15.88	15.81	15.81	16.01	15.81	16.14	16.41
525	7	8.26	9.61	8.26	8.26	8.36	8.26	8.63	9.37
<i>Different cardinalities</i>									
<i>Equal variances</i>									
105	3	5.01	5.01	5.01	5.01	5.01	5.01	6.38	5.01
105	7	5.55	7.62	7.77	11.72	6.77	6.75	10.26	5.55
525	3	5.67	5.67	5.70	5.67	5.67	5.67	5.89	5.67
525	7	5.65	7.49	8.11	10.71	6.89	6.35	8.48	5.66
<i>Different variances</i>									
105	3	11.73	11.77	11.73	11.73	11.73	11.73	12.23	11.86
105	7	7.63	9.77	8.65	9.82	8.37	7.68	9.76	7.69
525	3	13.82	13.84	13.87	13.82	13.82	13.82	14.12	14.2
525	7	7.46	9.18	9.76	10.17	8.53	7.72	9.27	8



CIMPA-UCR

Parametros

Notación: n es la cantidad de individuos
 k es la cantidad de clases

SA

Parámetro	Valor
Tasa de aceptación inicial χ_0	95%
Tasa de enfriamiento γ	90%
Largo de las cadenas de Markov L_c	$100 * n * (k - 1)$
Límite para detener las cadenas de Markov $nLimit$	$10 * n * (k - 1)$
Número de iteraciones del valor de la temperatura $nTemp$	150
Número máximo de temperaturas sin aplicar la regla de Metropolis $ncad$	4

TS

Parámetro	Valor
Longitud de la lista tabú $ T $	7
Cantidad máxima de iteraciones	$= n * k * T $

GA

Parámetro	Valor
Tamaño de la población $T_{population}$	40
Probabilidad de cruzamiento forzado p_c	0.25
Probabilidad de mutación p_m	0.001
Número de iteraciones	tamaño de la población * $k * \log(n)$
Frecuencia para aplicar nubes dinámicas	cada 10 iteraciones
Tipos de cruzamiento	cruzamiento forzado
Tipos de mutaciones	mutación simple

Resultados – Tiempo

Tiempo aproximado en minutos para cada corrida

Tabla	SS	BT	AG	Tabla	SS	BT	AG
AP0105cd03od06-e01	0.03	0.10	0.25	AP0525cd03od06-e01	0.13	5.00	1.00
AP0105cd03oi06-e01	0.02	0.09	0.25	AP0525cd03oi06-e01	0.13	5.00	1.00
AP0105cd07od06-e01	0.50	0.50	2.00	AP0525cd07od06-e01	0.50	16.00	1.60
AP0105cd07oi06-e01	0.33	0.50	0.67	AP0525cd07oi06-e01	2.00	15.00	1.00
AP0105ci03od06-e01	0.02	0.11	0.25	AP0525ci03od06-e01	1.33	5.00	1.00
AP0105ci03oi06-e01	0.02	0.10	0.67	AP0525ci03oi06-e01	0.33	2.75	0.40
AP0105ci07od06-e01	0.06	0.50	2.00	AP0525ci07od06-e01	1.25	12.00	4.00
AP0105ci07oi06-e01	0.05	0.50	1.50	AP0525ci07oi06-e01	0.31	9.75	3.00

Observaciones finales

- El método jerárquico de Ward es el peor
- Cuando las cardinalidades son iguales, los métodos obtienen los mismos óptimos; sin embargo, SS tiene mejor tasa de atracción (100%)
- Cuando las cardinalidades son diferentes y se desean muchas clases, SS y BT tienen problemas, incluso k-medias es mejor. En este caso, AGA y ACO son claramente los mejores
- ACO se comporta bien con diferentes varianzas y cardinalidades
- El método más rápido es k-medias y el más lento es BT
- La calibración de los parámetros es una tarea difícil.

