

# Introducción a los Algoritmos Genéticos

Francisco José Ribadas Pena, Santiago Fernández Lanza

INTELIGENCIA ARTIFICIAL

5 Informática

ribadas@uvigo.es, sflanza@uvigo.es

18 de marzo de 2013

# 1 Introducción

- Métodos de  $\left. \begin{array}{l} \text{búsqueda} \\ \text{aprendizaje} \\ \text{optimización} \end{array} \right\}$  que emulan de forma simplificada los principios de la evolución biológica
  - *los individuos mejor adaptados tienen mayores posibilidades de supervivencia y de reproducirse*
  - *los descendientes combinan características heredadas de sus progenitores*
  - *eventualmente se producen cambios aleatorios (mutaciones) en las características de algunos individuos (pueden favorecerlos o perjudicarlos)*
- La idea básica es partir de un conjunto de individuos (*población*) y aplicar sobre ellos una versión simplificada de las reglas anteriores.
  - cada individuo está caracterizado por determinada información genética (*genes*) que codifica datos de interés para el problema que se pretende resolver
  - cada individuo es evaluado por una función de evaluación (*fitness*)
    - "mide" su grado adaptación al medio
    - cómo de buena es la info. que codifica para el problema en cuestión
  - los individuos se cruzan entre sí para dar lugar a nuevos individuos que formarán parte de la siguiente *generación*
    - la probab. de selección se basa en el valor de la función *fitness*
  - una parte de los individuos de la generación actual sobrevivirá y pasará a la siguiente
    - la probab. de sobrevivir se basa en el valor de la función *fitness*
  - de vez en cuando algún individuo sufrirá un cambio aleatorio en su info. genética
  - después de varias (muchas) generaciones la población habrá convergido a un conj. "homogéneo" de individuos con evaluación alta (la solución buscada)

## 2 Codificación de info. genética y función fitness

Son los puntos que hay que definir a la hora de aplicar los algoritmos genéticos a un problema determinado.

### Codificación de la información genética

- Se trata de representar aquella información relevante del problema cuyos valores se desean estimar
- En su forma más simple se usa una representación binaria.
  - para cada uno de los atributos relevantes sus valores se codifican en binario usando  $k$  bits (forman un *gen*)
  - la información genética de cada individuo tiene la forma de una cadena de 0's y 1's de tamaño fijo
  - la posición de cada gen en la cadena es fija
- Ejemplo: codificación de una aleación de 4 metales para turbina de un reactor

Individuo i:	0001	1010	101	01101
	% Fe	% Cu	% Al	% Cd
- Otras alternativas: valores numéricos, árboles abstractos de análisis sintáctico

### Función fitness

- Responsable de evaluar a los individuos (valores más altos son mejores)
- Es completamente dependiente del problema
- Punto clave: determina lo buenos o malos que serán los resultados y la convergencia o no del método
- Decodifica la información genética y "puntuá" al individuo en función de ella
- Ejemplo: turbina reactor

*fitness*:  $8 \cdot \text{estimación del punto de fusión de la aleación} + 45 \cdot \text{estimación del índice de fricción con el aire}$

# 3 Operadores genéticos

## Proceso de selección

- Proceso utilizado para decidir qué individuos sobrevivirán y cuáles se reproducirán
- La probabilidad de selección debe de ser proporcional al valor *fitness* del individuo candidato

$$P(h_k \text{ sea seleccionado}) = \frac{fitness(h_k)}{\sum_{i=1}^{|Poblacion|} fitness(h_i)}$$

- El método más común es la *selección por ruleta* (ver pseudocódigo)
- Otra alternativa es la *selección por torneo*

## Operador de cruce

- Encargado de combinar la info. genética de 2 padres para dar lugar a 2 hijos
- Método más común: *cruce puntual*
  - Dadas las cadenas de bits del padre y de la madre
  - Se elige un *punto de cruce* al azar en la cadena genética
  - Se truncan las cadenas de los padres por ese punto
  - Se crean dos nuevos individuos, cada uno con una parte distinta de sus padres.

PADRES	HIJOS
0 0 0 1 1 0 1 0 1 0 1 0 1 1 0 1	0 0 0 1 1 1 0 0 0 0 1 0 1 0 1 0
1 0 1 0 1 1 0 0 0 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1
~	
pto. cruce	

- Otras alternativas: cruce en 2 puntos, cruce uniforme (patrón aleatorio de cruce)

## Operador de mutación

- Encargado de realizar la mutación en la info. de un individuo
- Muy poca frecuencia de ocurrencia (fenómeno raro)
  - puede no usarse
- Si en la generación actual va a producirse una mutación
  - Seleccionar al azar un individuo cualquiera (no selección por *fitness*)
  - Se elige un *punto de mutación* al azar en su cadena genética
  - Cambiar el valor del bit afectado
  - Devolver al nuevo individuo a la población

INDIV, ORIGINAL	0	0	0	1	1	0	1	0	1	0	1	1	0	1
INDIV. MUTADOO	0	0	0	1	1	0	1	0	1	1	1	1	0	1

^  
pto. mutacion

- La mutación sirve como mecanismo de control del proceso
  - tasa mutación alta: ayuda a evitar máximos locales
  - tasa mutación baja: facilita la convergencia

## 4 Algoritmo genérico

### Parámetros:

$n$ :	tamaño de la población (num. individuos)
$r$ :	tasa de reproducción (fracción de parejas que se cruzan)
$m$ :	tasa de mutación (prob. de que haya mutación en pobl. actual)
$max$ :	num. máximo de poblaciones
$fitness()$ :	función de evaluación

### Algoritmo

Generar una población  $P_0$  de  $n$  individuos aleatorios

$p = 0$  /\* Contador población \*/

$P_{actual} = P_0$

Bucle mientras  $p < max$

Para cada individuo  $i$ , calcular  $fitness(i)$

/\* Construir nueva población ( $P_{p+1}$ ) \*/

**Seleccionar**  $(2 * r) * n$  individuos, **cruzarlos** y añadir sus hijos a  $P_{p+1}$ .

**Seleccionar**  $(1 - 2 * r) * n$  individuos y añadirlos a  $P_{p+1}$ .

Si  $toca\_mutacion(m)$

Elegir un individuo de  $P_{p+1}$  al azar

Aplicarle el operador de **mutación** y devolverlo a  $P_{p+1}$

$P_{actual} = P_{p+1}$

$p = p + 1$

Fin bucle

Pseudocódigo *Método de selección por ruleta.*

Calcular **S**, la suma de todos los valores  $fitness$  de la población actual

Generar un  $n^o$  aleatorio, **A**, entre **0** y **S**

Recorrer los individuos de la población acumulando su valores de  $fitness$

Si el valor acumulado es **mayor o igual** que **A**, devolver individuo actual