

Inteligencia Artificial

Guía 5.2. Algoritmos Genéticos

4° Licenciatura En Sistemas de Información

2020

Universidad: UADER FCyT Concepción del Uruguay

Profesor: Lopez De Luise Daniela, Bel Walter

Alumnos: Exequiel Gonzalez, Cepeda Leandro

1. Dados los dígitos 0 a 9 y los operadores +, -, *, / Hallar una secuencia de operaciones simples que resulten y representen dicho número. Los operadores deben aplicarse secuencialmente de izquierda a derecha a medida que se lee la ecuación.

Para el número 23
Una secuencia solución es $6+5*4/2+1$

Para el número 75.5
La posible solución es $5/2+9*7-5$

0: 0000
1: 0001
2: 0010
3: 0011
4: 0100
5: 0101
6: 0110
7: 0111
8: 1000
9: 1001
+: 1010
-: 1011
*: 1100
/: 1101

****Los genes 1110 y 1111 no se usarán de momento (se ignorarán cada vez que el algoritmo los encuentre)
Considere que el gen pueda contener hasta 9 códigos (dígitos u operaciones).
TIP: ignoramos las expresiones que no respeten el patrón: # -> operador -> # -> operador ...**

a) Cuántos genes se necesitan?

Se necesitan 9 genes como máximo. Es decir 36 bits en total, ya que $4\text{bits/gen} \times 9\text{gen} = 36\text{bits}$

b) De al menos dos códigos de 23

$23 = 4+5*4/2+5 = 010010100101110001001101001010100101$

0100	1010	0101	1100	0100	1101	0010	1010	0101
4	+	5	*	4	/	2	+	5

$23 = 2*5*3-7 = 00101100010111000011101101111111111$

0010	1100	0101	1100	0011	1011	0111	1111	1111
2	*	5	*	3	-	7	n/a	n/a

c) Determine al menos dos funciones de fitness para el problema

$f = 1/(\text{solucion-target})$
 $f = 1/(42-23) = 1/19$
 $f = 1/(42-37) = 1/5$

d) Hacer el algoritmo/seudocódigo:

1. entrar una cadena
2. parsear las cadenas de bits en valores
3. hallar los códigos válidos y reemplazar los bits por los símbolos
4. descartar las si hay 'n/a'
5. evaluar fitness

ALGORITMO algoritmo_genetico

AMBITO

valoresValidos = [

'0000': 0,
'0001': 1,
'0010': 2,
'0011': 3,
'0100': 4,
'0101': 5,
'0110': 6,
'0111': 7,
'1000': 8,
'1001': 9,
'1010': '+',
'1011': '-',
'1100': '*',
'1101': '/' : registro

cadena : string

cromosoma, cromosomaParseado : array[1..n]

target, solucion, fitness : real

```

Funcion cromosoma(cadena : string) : vector[1..n]
Var i,j : entero;
Inicio
|   j ← 0
|   Para i : 1 ... tamaño(cadena), i=i+4 hacer
|   |   Si i <> 36 entonces
|   |   |   j ← j + 1
|   |   cromosoma[j] ← cadena[i:i+4]
|   |   Fin si
|   Fin para
Fin

```

```

Funcion parsearCromosoma(cromosoma : vector[1..n], valoresValidos : registro) : vector[1..n]
Var i; entero; gen: string
Inicio
|   Para i : 1 .. tamaño(cromosoma) hacer
|   |   Si cromosoma[i] esta en registro entonces
|   |   |   gen ← valoresValidos.cromosoma[i]
|   |   |   parsearCromosoma[i] ← gen
|   |   Fin si
|   Fin para
Fin

```

```

Funcion target(cromosomaParseado : vector[1..n]) : real
Var i; entero; a: real; op: char
Inicio
|   target ← cromosomaParseado[0]
|   Para i : 1 .. tamaño(cromosomaParseado), i=i+2 hacer
|   |   Si cromosoma[i] esta en registro entonces
|   |   |   a ← cromosoma[i+1]
|   |   |   op ← cromosoma[i]
|   |   |   Segun op hacer
|   |   |   |   op='+' hacer target = target + a
|   |   |   |   op='- ' hacer target = target - a
|   |   |   |   op='*' hacer target = target * a
|   |   |   |   op='/' hacer target = target / a
|   |   |   Fin segun
|   |   Fin si
|   Fin para
Fin

```

```

Funcion fitness(solucion : real, target : real) : real
Inicio
|   fitness ← 1/(solucion-target)
Fin

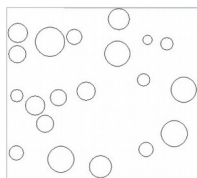
```

```

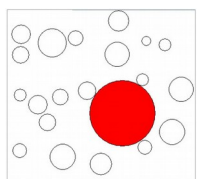
CUERPO
Inicio
|   Leer(cadena)
|   Leer(solucion)
|   cromosoma ← cromosoma(cadena)
|   cromosomaParseado ← parsearCromosoma(cromosoma, valoresValidos)
|   target ← target(cromosomaParseado)
|   fitness ← fitness(solucion, target)
|   Mostrar("Cromosoma: ", cromosoma)
|   Mostrar("Cromosoma parseado: ", cromosomaParseado)
|   Mostrar("Target: ", target)
|   Mostrar("Fitness: ", fitness)
Fin

```

2) Dado el área como un número correspondiente a círculos dispersos y que no se superponen dentro de una superficie como en la figura:



- Determine cómo codifica en genes de un AG para encontrar el disco de radio más grande que puede caber entre estos discos, sin que se superpongan entre sí.
- Determine la función de fitness para este problema. El resultado debe dar por ejemplo algo como el disco rojo de la figura :



- Determine la función de cruce que sea compatible.
- Determine una función de mutación que sea compatible.

Planteamos el problema como una matriz de $n \times m$ que representa el espacio total donde se ubicaran los círculos.

Donde $n, m \geq 1 \wedge n, m \in \mathbb{Z}^+$

Cada **círculo** se representará como un área de n^2 .

Cada elemento de la matriz, corresponde a un **gen** y se representa de la forma **xij**, donde **i =fila**, **j=columna**, con las siguientes restricciones:

xij = 1 → hay una parte del área del círculo en la posición ij

xij = 0 → no hay una parte del área del círculo en la posición ij

Ejemplo: para la imagen del círculo rojo, si lo dividiéramos en 4 cuadrantes y tomamos el cuadrante inferior derecho, podría representarse de la siguiente manera:

cromosoma =

0₁₁ 0₁₂ 0₁₃ 1₁₄ 0₁₅ 0₁₆ 1₁₇ 1₁₈ 1₂₁ 1₂₂ 1₂₃ 1₂₄ 1₂₅ 0₂₆ 1₂₇ 1₂₈ 0₃₁ 1₃₂ 1₃₃ 1₃₄ 1₃₅ 0₃₆ 0₃₇ 0₃₈ 0₄₁ 1₄₂ 1₄₃ 1₄₄ 1₄₅ 0₄₆ 0₄₇ 0₄₈ 0₅₁ 1₅₂ 1₅₃ 1₅₄

1₅₅ 1₅₆ 1₅₇ 0₅₈ 0₆₁ 0₆₂ 0₆₃ 1₆₄ 0₆₅ 1₆₆ 1₆₇ 0₆₈ 1₇₁ 1₇₂ 0₇₃ 0₇₄ 0₇₅ 0₇₆ 0₇₇ 0₇₈ 1₈₁ 1₈₂ 0₈₃ 0₈₄ 0₈₅ 0₈₆ 0₈₇ 0₈₈

			1			1	1
1	1	1	1	1		1	1
	1	1	1	1			
	1	1	1	1			
	1	1	1	1	1	1	
			1		1	1	
1	1						
1	1						

Funcion fitness: esta representada por $1/(\text{area total} - \text{area del círculo})$

Ejemplo: continuando en el ejemplo anterior, la función fitness estaría dada por:
 $f = 1/(64 - 16) = 0,020833333 \Rightarrow 1/48$

Podríamos decir que el área de mayor tamaño que podría abarcar un círculo se regiría según las siguiente restricciones:

Si $n=m$ la función de fitness seria: $f = 1/(n*m) - (n^2)$
 Si $n > m$ la función de fitness seria: $f = 1/(n*m) - (m^2)$
 Si $n < m$ la función de fitness seria: $f = 1/(n*m) - (n^2)$

Función de cruza

madre:

0₁₁ 0₁₂ 0₁₃ 1₁₄ 0₁₅ 0₁₆ 1₁₇ 1₁₈ 1₂₁ 0₂₂ 0₂₃ 0₂₄ 0₂₅ 0₂₆ 1₂₇ 1₂₈ 1₃₁ 1₃₂ 1₃₃ 0₃₄ 0₃₅ 0₃₆ 0₃₇ 0₃₈ 1₄₁ 1₄₂ 1₄₃ 0₄₄ 0₄₅ 0₄₆ 0₄₇ 0₄₈ 1₅₁ 1₅₂ 1₅₃ 0₅₄

0₅₅ 1₅₆ 1₅₇ 0₅₈ 0₆₁ 0₆₂ 0₆₃ 1₆₄ 0₆₅ 1₆₆ 1₆₇ 0₆₈ 1₇₁ 1₇₂ 0₇₃ 0₇₄ 0₇₅ 0₇₆ 0₇₇ 0₇₈ 1₈₁ 1₈₂ 0₈₃ 0₈₄ 0₈₅ 0₈₆ 0₈₇ 0₈₈

			1			1	1
1						1	1
1	1	1					
1	1	1					
1	1	1			1	1	
			1		1	1	
1	1						
1	1						

Cruza:

$0_{11} 0_{12} 0_{13} 1_{14} 0_{15} 0_{16} 1_{17} 1_{18} 1_{21} 1_{22} 1_{23} 1_{24} 1_{25} 0_{26} 1_{27} 1_{28} 0_{31} 1_{32} 1_{33} 1_{34} 1_{35} 0_{36} 0_{37} 0_{38} 0_{41} 1_{42} 1_{43} 1_{44} 1_{45} 0_{46} 0_{47} 0_{48} 1_{51} 1_{52} 1_{53} 0_{54}$
 $0_{55} 1_{56} 1_{57} 0_{58} 0_{61} 0_{62} 0_{63} 1_{64} 0_{65} 1_{66} 1_{67} 0_{68} 1_{71} 1_{72} 0_{73} 0_{74} 0_{75} 0_{76} 0_{77} 0_{78} 1_{81} 1_{82} 0_{83} 0_{84} 0_{85} 0_{86} 0_{87} 0_{88}$

			1		1	1	1
1	1	1	1	1		1	1
	1	1	1	1			
	1	1	1	1			
1	1	1			1	1	
			1		1	1	
1	1						
1	1						

Mutación:

$0_{11} 0_{12} 0_{13} 1_{14} 0_{15} 0_{16} 1_{17} 1_{18} 1_{21} 1_{22} 1_{23} 1_{24} 1_{25} 0_{26} 1_{27} 1_{28} 0_{31} 1_{32} 0_{33} 1_{34} 1_{35} 0_{36} 1_{37} 0_{38} 0_{41} 1_{42} 1_{43} 1_{44} 1_{45} 0_{46} 0_{47} 0_{48} 1_{51} 0_{52} 1_{53} 0_{54}$
 $0_{55} 1_{56} 1_{57} 0_{58} 0_{61} 0_{62} 0_{63} 1_{64} 1_{65} 1_{66} 0_{67} 0_{68} 1_{71} 1_{72} 0_{73} 0_{74} 0_{75} 0_{76} 0_{77} 0_{78} 1_{81} 1_{82} 0_{83} 0_{84} 0_{85} 0_{86} 0_{87} 0_{88}$

			1		1	1	1
1	1	1	1	1		1	1
	1	0	1	1		1	
	1	1	1	1			
1	0	1			1	1	
			1	1	1	0	
1	1						
1	1						