

Maestría y Doctorado en Ciencia de la Computación

Inteligencia Artificial

Programación Genética

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe
22 de Agosto del 2020



UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

Índice



Objetivos del Curso

Computación Evolutiva

Programación Genética

Objetivos del Curso



- ▶ **Conocer, comprender e implementar algoritmos evolutivos para resolver problemas complejos.**
- ▶ Conocer, comprender e implementar algoritmos de inteligencia de enjambre para resolver problemas complejos.
- ▶ Conocer, comprender e implementar algoritmos inmunes artificiales para resolver problemas complejos.
- ▶ Conocer, comprender e implementar sistemas basados en lógica difusa para resolver problemas complejos.

Computación Evolutiva

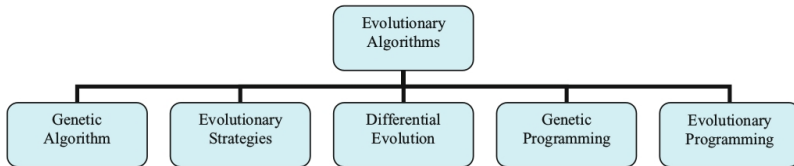


Figure: Algoritmos Evolutivos

Programación Genética



- ▶ Uno de los principales problemas en ciencia de la computación es como hacer que las computadoras resuelvan problemas sin ser explícitamente programadas para ello.
- ▶ La Programación Genética ofrece una solución mediante la evolución de programas de computadoras por medio de métodos de selección natural.

Programación Genética

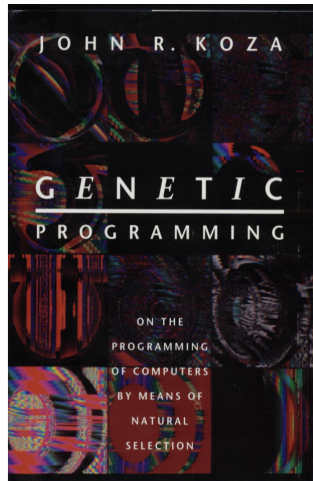


- ▶ En 1975, John Holland creó los Algoritmos Genéticos [1].
- ▶ En 1992, John Koza [2] usó los Algoritmos Genéticos para desarrollar programas de computadora que realizan ciertas tareas. Llamó a su método de Programación Genética (PG).

Programación Genética



© Copyright 2004, Cgen Research Corporation





Programación Genética

- Programación: Es la forma de indicarle a una computadora lo que tiene que hacer mediante un conjunto de instrucciones. Podemos representar un programa mediante un árbol.

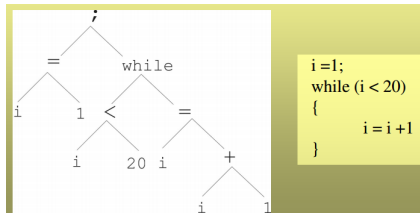


Figure: Algoritmos Evolutivos

- Genética: Utiliza principios de selección natural.

Programación Genética



- ▶ La PG es una extensión de los AG en el dominio de los programas, donde:
 - ▶ El individuo es un programa de computador.
 - ▶ El espacio de búsqueda son todos los posibles programas de computador.
- ▶ En resumen, la PG es un método de búsqueda, dentro de un espacio significativamente polinomial/exponencial y restringido de programas de computador, que provee como resultado una solución (exacta o por lo menos aproximada) para resolver determinado problema.

Programación Genética

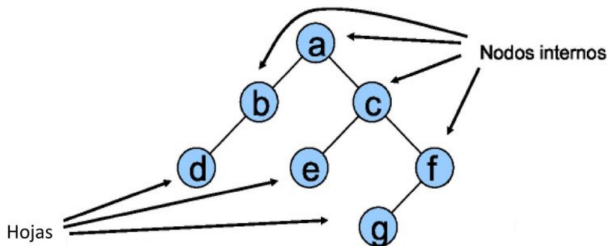


- ▶ Un programa de computador es una extensión matemática compuesta de funciones y terminales.
- ▶ Las funciones pueden ser operaciones aritméticas (+, -, *, ...), operaciones booleanas (and, or, not, ...), funciones matemáticas (sen, cos, ...), operadores condicionales (if, then, ...), funciones de iteración (for, while, ...), funciones de recursión, funciones específicas del problema, entre otras.
- ▶ Los terminales pueden ser variables (representando posiblemente, las entradas) o constantes (número 5.32).



Programación Genética

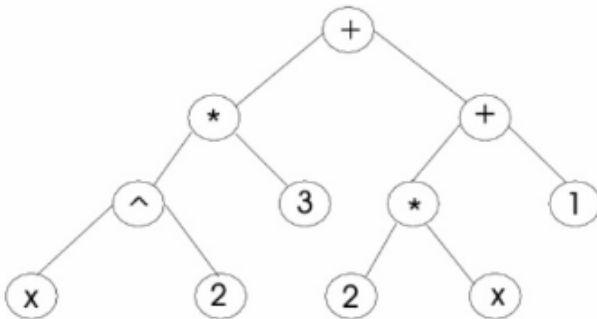
- ▶ Los programas de computador en PG fueron representados en forma de árboles.
- ▶ Las funciones aparecen en los nodos internos del árbol.
- ▶ Los terminales aparecen en los nodos hoja.





Programación Genética - Ejemplo Representación de Árbol

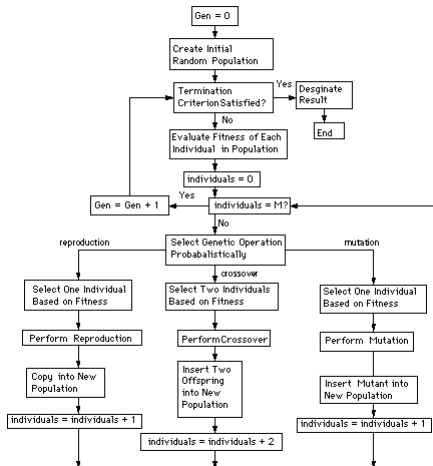
$$3x^2+2x+1$$





Programación Genética - Flujograma

Flowchart for Genetic Programming



Programación Genética - Representación de Individuos



- ▶ Antes de definir la representación de los individuos, debemos definir el conjunto de funciones y terminales.
- ▶ Por ejemplo:
 - ▶ Un conjunto de funciones que contiene los operadores matemáticos suma (+), resta (-), multiplicación (*) y división (/).
 - ▶ Conjunto de terminales que contiene dos variables X , y un conjunto constante que contiene los valores enteros -5, -4, -3, -2, -1, 1, 2, 3, 4, 5.

Programación Genética - Representación de Individuos



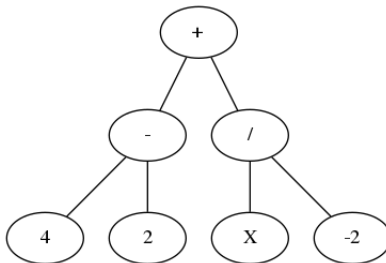
- ▶ Una vez definido el conjunto de funciones y terminales, podemos definir diferentes formas de representación de individuos.
- ▶ Vamos a definir una representación tipo árbol estática. Dicha representación la usaremos para representar un árbol conformado por 7 funciones y/o terminales.
- ▶ Por ejemplo:



Programación Genética - Representación de Individuos 7

4	-	2	+	X	/	-2
---	---	---	---	---	---	----

Candidate : $(4 - 2) + (\frac{X}{-2})$



Programación Genética - Aptitud



16

- ▶ Cada individuo tiene asociado a si una medida numérica (aptitud), que es el resultado de la interacción con el ambiente, es decir, es una medida de grado de adaptación del individuo. Está relacionada al proceso evolutivo, haciendo que tenga mayor probabilidad de que sus características sean propagadas y permanecer en las siguientes poblaciones.
- ▶ Por ejemplo: Encontrar una función matemática que se ajuste al siguiente conjunto de entradas y salidas:

Input	Output
10	29
42	125
3	8

Programación Genética - Aptitud



- ▶ Existen diferentes formas de definir la aptitud de un individuo.
- ▶ Vamos a utilizar la función de Error Cuadrático Medio (EMC) el cual mide el promedio de los errores al cuadrado, es decir, la diferencia entre el valor esperado menos y lo que se estima, dividido entre la cantidad de comparaciones.

$$EMC = \frac{1}{n} \sum_1^n \left(\hat{Y}_i - Y_i \right)^2$$

- ▶ Cuanto menor sea el valor de la aptitud, mejor será el candidato.



Programación Genética - Aptitud

- Considere el siguiente individuo:

3	+	2	+	1	*	1
---	---	---	---	---	---	---

Candidate 1 : $(3 + 2) + (1 * 1)$

Input	Expected	Actual	Difference
10	29	6	23
42	125	6	119
3	8	6	2

$$Fitness = \frac{23^2 + 119^2 + 2^2}{3} = 4898.000$$



Programación Genética - Aptitud

- Considere el siguiente individuo:

X	-	X	+	X	+	X
---	---	---	---	---	---	---

Candidate 2 : $(X - X) + (X + X)$

Input	Expected	Actual	Difference
10	29	20	9
42	125	84	41
3	8	6	2

$$Fitness = \frac{9^2 + 41^2 + 2^2}{3} = 588.667$$

Programación Genética - Aptitud



- Para los dos individuos:

Rank	Candidate	Fitness
1st	Candidate 1	588.667
2nd	Candidate 2	4898.000

Programación Genética

Operadores Genéticos



- ▶ Podemos considerar las siguientes probabilidades para los operadores genéticos:
 - ▶ Reproducción o replicación (entre el 10% al 20%)
 - ▶ Cruzamiento (entre el 40% al 70%)
 - ▶ Mutación (entre el 20% al 50%)

Programación Genética - Cruzamiento

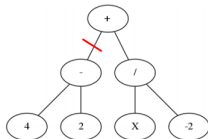


- ▶ Antes del cruzamiento debemos seleccionar 2 individuos (torneo o ruleta).
- ▶ Podemos utilizar los cruzamientos visto para la representación binaria en los AG:
 - ▶ Cruzamiento de 1 punto.
 - ▶ Cruzamiento de n puntos.
 - ▶ Cruzamiento uniforme.
- ▶ Obtenemos 2 descendientes.

PG - Cruzamiento de 1 punto (Punto 3)

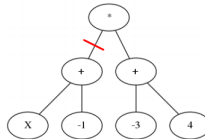
4 - 2 + X / -2

$$(4 - 2) + \left(\frac{X}{-2}\right)$$



X + -1 * -3 + 4

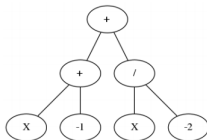
$$(X + -1) * (-3 + 4)$$



Cruzamiento 1 punto (Punto 3)

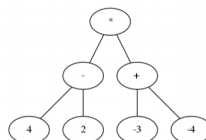
X + -1 + X / -2

$$(X + -1) + \left(\frac{X}{-2}\right)$$



4 - 2 * -3 + 4

$$(4 - 2) * (-3 + 4)$$





Programación Genética - Mutación

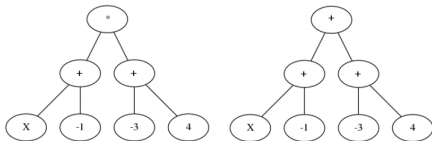
- ▶ Antes de la mutación debemos seleccionar a un individuo aleatoriamente.
- ▶ Podemos utilizar la mutación simple, donde es seleccionado una función o terminal de forma aleatoria y modificada por otra función o terminal respectivamente. Por ejemplo:

X	+	-1	*	-3	+	4
---	---	----	---	----	---	---

$$(X + -1) * (-3 + 4)$$

X	+	-1	+	-3	+	4
---	---	----	---	----	---	---

$$(X + -1) + (-3 + 4)$$



- ▶ Importante: Manejar los problemas de resultados no válidos de las operaciones.



Ejemplo de Implementación de un algoritmo de PG

- Encontrar una función matemática que se ajuste al siguiente conjunto de entradas y salidas:

Input	Output
0	0
0.1	0.005
0.2	0.02
0.3	0.045
0.4	0.08
0.5	0.125
0.6	0.18
0.7	0.245
0.8	0.32
0.9	0.405

Ejemplo de Implementación de un algoritmo de PG



- ▶ Considere los siguientes parámetros:
 - ▶ Tamaño de la población: 8 individuos.
 - ▶ Cantidad de genes: 7 .
 - ▶ Funciones: $+$, $-$, $*$, $/$
 - ▶ Terminales (Constantes): $-5, -4, -3, -2, -1, 1, 2, 3, 4, 5$
 - ▶ Terminales (Variables): X
 - ▶ Probabilidad de Reproducción: 20%
 - ▶ Selección para Reproducción: Torneo 3
 - ▶ Probabilidad de Cruzamiento: 40%
 - ▶ Selección para Cruzamiento: Torneo 2
 - ▶ Cruzamiento de un Punto (Punto Aleatorio)

Ejemplo de Implementación de un algoritmo de PG



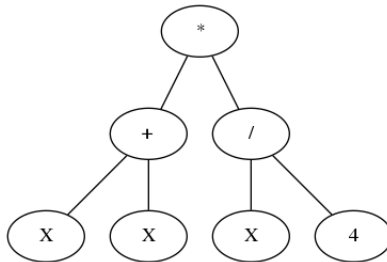
- ▶ Considere los siguientes parámetros:
 - ▶ En el caso de que solo quede un espacio en la nueva población y se debe realizar un Cruzamiento, escoger aleatoriamente un descendiente para la nueva población.
 - ▶ Probabilidad de Mutación: 40%
 - ▶ Selección para Mutación: Torneo 3
 - ▶ Mutación simple.
 - ▶ Cantidad de Interacciones: 50



Ejemplo de Implementación de un algoritmo de PG

X	+	X	*	X	/	4
---	---	---	---	---	---	---

$$(X + X) * \frac{X}{4} = 2X * \frac{X}{4} = \frac{2X^2}{4} = \frac{X^2}{2}$$



¡GRACIAS!



Bibliografía



- [1] J. H. Holland.
Adaptation in Natural and Artificial Systems.
University of Michigan Press, Ann Arbor, MI, 1975.
- [2] J. R. Koza.
Genetic Programming: On the Programming of Computers by Means of Natural Selection.
MIT Press, Cambridge, MA, USA, 1992.