



## Conceptos básicos de programación genética

José Jesús Martínez Páez, Profesor, facultad de ingeniería, Universidad Nacional de Colombia, josej@ingenieria.ingsala.unal.edu.co

### INTRODUCCIÓN

La Programación Genética, PG, es un retoño de los Algoritmos Genéticos, en la cual los cromosomas que sufren la adaptación son en sí mismos programas de computador. Se usan operadores genéticos especializados que generalizan la recombinación sexual y la mutación, para los programas de computador estructurados en árbol que están bajo adaptación.

La PG trata de resolver uno de las cuestiones más excitantes e interesantes de las ciencias de la computación: ¿cómo pueden aprender los computadores a solucionar problemas sin que se les programe explícitamente? En otras palabras, la cuestión es cómo podemos hacer para que los computadores hagan lo que tienen que hacer, sin necesidad de la intervención humana que les diga exactamente como lo deben hacer.

El espacio de búsqueda de la PG es el espacio de todos los posibles programas de computador compuestos de funciones y elementos terminales apropiados al dominio del problema. Las funciones pueden ser operaciones aritméticas, operaciones de programación, funciones matemáticas, funciones lógicas, y funciones específicas del dominio del problema.

Todo programa de computador se puede considerar como un árbol. Supongamos que tenemos un programa sencillo como puede ser calcular en grados centígrados una temperatura dada en grados Fahrenheit.

Se tiene que la fórmula es  $C = 5/9 F - 32$ . En la figura 1. se presenta el árbol que representa el programa que convierte grados Fahrenheit en grados centígrados. Los nodos del árbol corresponden a las operaciones matemáticas, que en otros casos pueden ser funciones, que se efectúan entre los operandos o terminales, que corresponden a las hojas del árbol. En este caso F representa la temperatura en grados Fahrenheit y C el resultado en grados centígrados.

### 2. Pasos preparatorios para solución de un problema por PG

En la aplicación de PG, hay cinco pasos preparatorios importantes:

1. Determinar el conjunto de terminales,
2. Determinar el conjunto de funciones primitivas,
3. Definir la medida de la aptitud,
4. Configurar los parámetros para controlar la corrida, y
5. Establecer el método para designar un resultado y el criterio de terminación de la ejecución del programa.

Vamos a seguir los pasos para el ejemplo de conversión de grados Fahrenheit en grados centígrados.

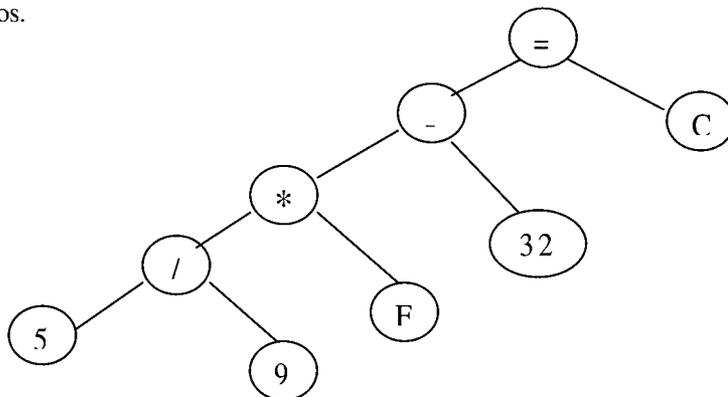


Figura 1. Árbol correspondiente al programa que hace la conversión de grados Fahrenheit a grados centígrados.

**Paso 1.** Consiste en identificar el conjunto de terminales. Los terminales son las hojas de los árboles, que corresponden a valores constantes. El conjunto de terminales puede incluir el conjunto de los números enteros, el de los números naturales, etc. que se pueden ver como las entradas al programa de computador que aún no se ha obtenido. El conjunto de terminales y el conjunto de funciones son los ingredientes a partir de los cuales, mediante

un proceso evolutivo artificial, la PG, trata de construir un programa de computador para solucionar, total o parcialmente, un problema.

En este caso vamos a definir el conjunto de terminales  $t$  que en este caso son dos conjuntos. Uno que corresponde a varios valores enteros dentro de los cuales vamos a escoger para colocar en las hojas: {4, 5, 7, 9, 30, 32, 34} y el conjunto {F} que corresponde al valor en grados Fahrenheit que quiero convertir a grados centígrados.

**Paso 2.** Consiste en identificar el conjunto de funciones  $f$  que se van a usar, para generar la expresión matemática que trata de satisfacer la muestra finita de datos. En este caso son las operaciones elementales: {-, \*, /}.

Cada programa de computador es entonces una composición de funciones del conjunto de funciones  $f$  y terminales del conjunto de terminales  $t$ . Dichas funciones deben poder aceptar, como sus argumentos, cualquier valor y tipo de dato que pueda retornar la función correspondiente del conjunto de funciones, y cualquier valor y tipo de dato que posiblemente, pueda tomar por cualquier terminal del conjunto de terminales. Esto es, el conjunto de funciones y el conjunto de terminales deben tener la propiedad de clausura.

La PG comienza con una población inicial de programas de computador compuestos de funciones y terminales apropiadas al dominio del problema. La creación de esta población inicial es, en efecto, una búsqueda aleatoria ciega, en el espacio de búsqueda del dominio del problema representado como programas de computador.

**Paso 3.** Cada programa individual, se mide en términos de que tan bien se comporta en el ambiente del problema particular. Como en el caso de los AG, esta medida se llama medida de aptitud y representa su idoneidad como candidato solucionador del problema. La naturaleza de la medida de aptitud varía con el problema. Para este caso podemos tener una tabla con algunos valores

equivalentes en ambas escalas. Podemos probar los programas generados contra esta escala y ver que también se comportan; es decir, que tan cercanos estén sus resultados a la tabla.

Para muchos problemas, la aptitud se mide por el error producido por el programa de computador. El mejor programa de computador tendrá un error cercano a cero. Si se están tratando ejemplos de reconocimiento de patrones o de clasificación, la aptitud de un programa particular se puede medir por una combinación del número de instancias manejadas correctamente (cierto positivo, y falso negativo) y número de instancias manejadas incorrectamente (cierto negativo, y falso positivo). Para algunos problemas, puede ser apropiado usar una medida de aptitud multiobjetivo, que incorpore una combinación de factores tales como corrección, parsimonia, o eficiencia.

Normalmente, cada programa de computador de la población se corre para varios casos de aptitud, de manera que su aptitud se mida como una suma o producto de una variedad de situaciones representativas diferentes. Los programas de computador de la población inicial, o generación 0, generalmente tienen una aptitud excesivamente pobre. No obstante, algunos individuos de la población serán más aptos que otros. Estas diferencias en el comportamiento son las que el programa genético se encarga de explorar posteriormente.

La operación reproducción incluye la selección de un programa de computador de la población actual de programas, con base en su aptitud, permitiéndole sobrevivir al producirlo en la nueva población. La operación cruce se usa para crear nuevos hijos programas de computador, a partir de dos programas padres seleccionados por su buena aptitud. Los programas padres en PG, normalmente son de diferentes tamaños y formas. Los programas hijos se componen de subexpresiones (subárboles, subprogramas, subrutinas, bloques) de sus padres. Así, los programas hijos normalmente son de diferentes tamaños y formas que sus padres, y además, posiblemente

serán más idóneos que sus progenitores.

Intuitivamente, si dos programas de computador tienen alguna efectividad en solucionar un problema, entonces alguna de sus partes debe tener algún mérito. La recombinación de partes escogidas aleatoriamente, de programas con alguna efectividad, a veces produce nuevos programas de computador que son aun más aptos para solucionar un problema determinado, que cualquiera de sus antecesores. La operación mutación también se utiliza en PG.

Después de que se efectúan las operaciones genéticas sobre la población actual, la población de hijos reemplaza la generación anterior, a cada individuo de la nueva población se le mide su aptitud, y el proceso se repite nuevamente, durante muchas generaciones. En cada etapa de este proceso altamente paralelo, controlado localmente, el estado del proceso consistirá únicamente de la población actual de individuos. La fuerza que controla este proceso consiste solo en la aptitud observada de los individuos en su lucha con el ambiente problema.

**Paso 4.** Se definen los parámetros de corrida como son el número de generaciones, el número de individuos programa o población, y la profundidad del árbol. Esta última característica es bien importante porque en operaciones del cruce el árbol puede aumentar de tamaño y se debe tomar una decisión ya que el no tomarla llevaría muy pronto a problemas de falta de memoria, además, se mejora el comportamiento de los árboles que sufren el proceso de evolución.

**Paso 5.** El mejor individuo que aparece en cualquier generación de una corrida, es decir el mejor hasta ahora, normalmente se designa como el resultado producido por la ejecución del programa genético. Sin embargo, en muchos casos el criterio de parada es el número de generaciones, dado que es bien difícil que en los primeros experimentos se logre una solución.

Una característica de la PG es la ausencia, o el papel relativamente mínimo, del preprocesamiento de entradas y el

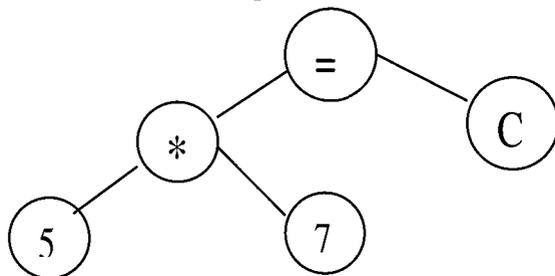
posprocesamiento de salidas. Las entradas, resultados intermedios, y salidas se expresan normalmente directamente en términos de la terminología natural del dominio del problema. El posprocesamiento de la salida del programa, si lo hay, se hace con una interfaz de salida.

Otra característica importante de la PG es que las estructuras que sufren la adaptación genética son estructuras activas. No son codificaciones pasivas (cromosomas) de la solución del problema. En su lugar, dado un sistema sobre el cual correr, las estructuras en PG son estructuras activas capaces de ser ejecutadas en su forma corriente.

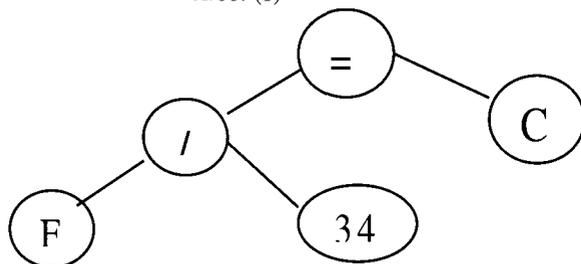
### 3. Búsqueda de programas por PG

La PG incuba programas de computador ejecutando los siguientes pasos:

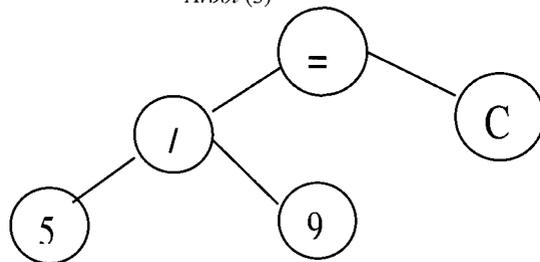
1. Genera una población inicial de composiciones aleatorias de funciones y terminales del problema.
2. Realiza repetidamente los siguientes pasos hasta que se satisfaga el criterio de terminación:
  - a. Ejecuta cada programa de la población y le asigna un valor de aptitud de según su comportamiento.
  - b. Crea una nueva población de programas aplicando las siguientes dos operaciones primarias, a los programas escogidos con una probabilidad basada en la aptitud.



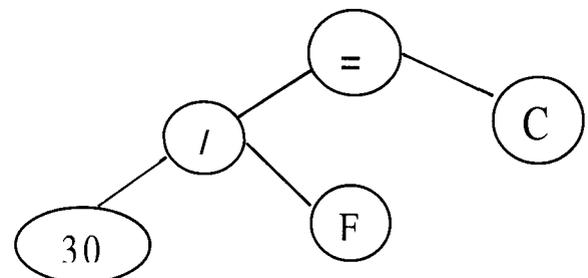
Árbol (1)



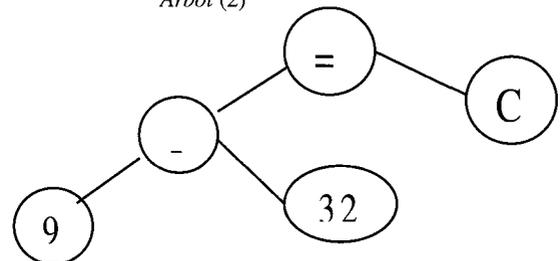
Árbol (3)



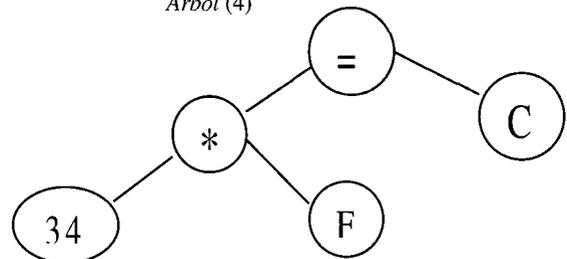
Árbol (5)



Árbol (2)



Árbol (4)



Árbol (6)

Figura 2. Población obtenida aleatoriamente del conjunto de funciones y de terminales.

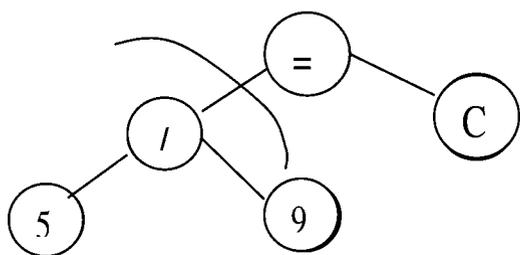
- Reproduce un programa existente copiándolo en la nueva población.
- Crea dos programas de computador a partir de dos programas existentes recombinao genéticamente partes escogidas de los dos programas en forma aleatoria, usando la operación cruce aplicada a un punto de cruce escogido aleatoriamente dentro de cada programa.

El programa con la mayor aptitud, se designa como el resultado de la corrida del programa genético. Este resultado representa una solución (o una solución aproximada) al problema. El cruce genético (recombinación sexual) opera sobre dos programas padres, seleccionados con una probabilidad basada en la aptitud y produce dos nuevos programas hijos consistentes de partes de cada padre.

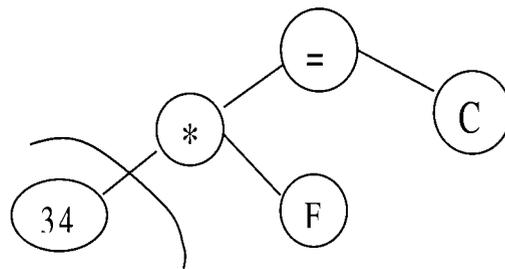
### 4. Obtención del programa de conversión

Vamos a seguir el algoritmo de la PG con nuestro ejemplo, suponiendo una población de seis árboles sencillos. Ver figura 2. Se tiene combinaciones de operaciones, o funciones, y terminales para obtener los árboles. Vamos a suponer que todos fueron evaluados con el mismo valor y se seleccionaron para ser padres de la siguiente generación. Ahora supongamos que se seleccionaron para cruce, o recombinación sexual los árboles 1-2, 3-4 y 5-6. También suponemos para el cruce los mismos puntos.

En la figura 3 se presenta el cruce de los árboles 5 y 6. En la figura 4, siguiendo el mismo procedimiento se presenta la población de hijos obtenida del cruce.

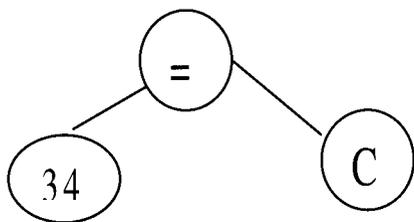


Árbol (5)

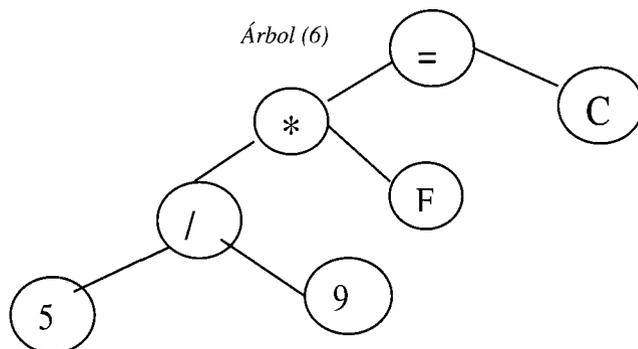


Árbol (6)

Figura 3. Cruce de los árboles 5 y 6.

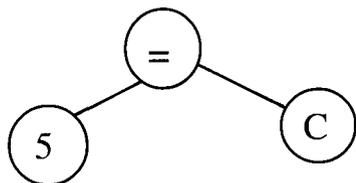


(e)

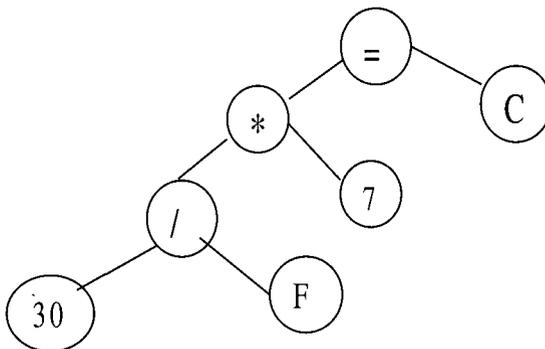


(f)

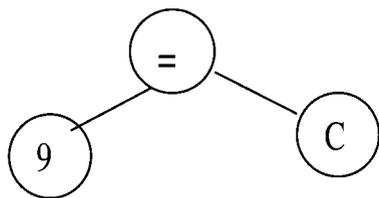
Figura 4. Cruce de los árboles 5 y 6, se obtienen los hijos e y f. En la figura 5 se presentan los árboles hijos, a y b de 1 y 2; c y d de 3 y 4.



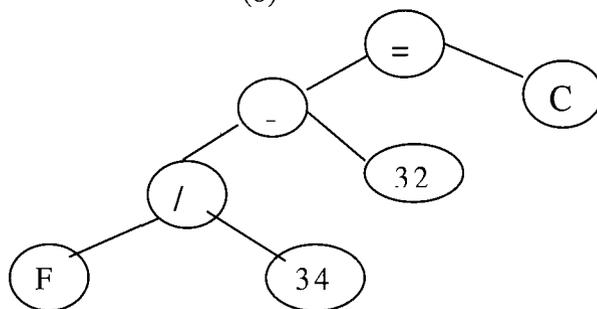
(a)



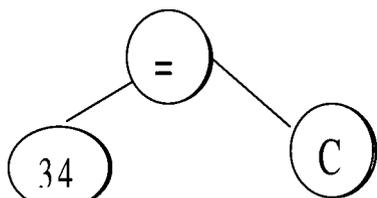
(b)



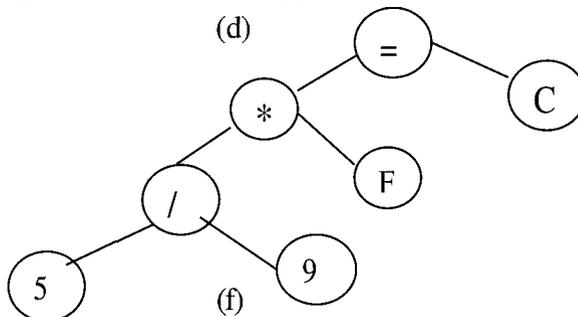
(c)



(d)



(e)



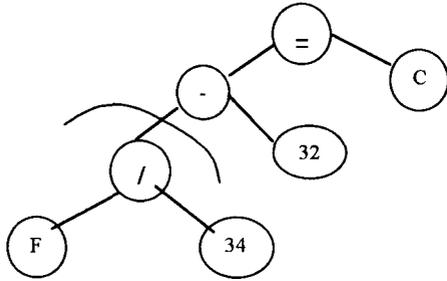
(f)

Figura 5. Árboles hijos del cruce de la primera generación.

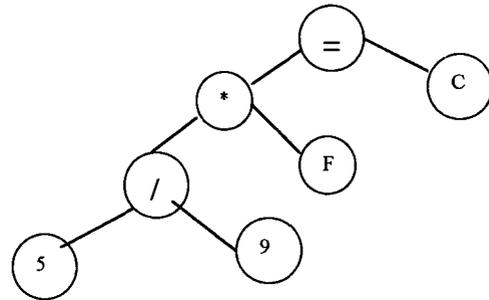
## CONCEPTOS BÁSICOS DE PROGRAMACIÓN GENÉTICA

Una vez obtenida esta población de hijos, se evalúan, con base en la cercanía del resultado. Vamos a suponer que la función de

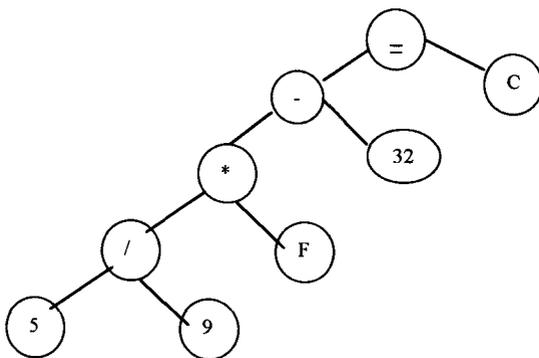
Ahora crucemos los árboles d y f. Se obtienen los árboles h e i.



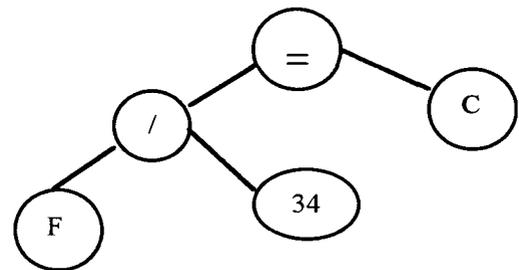
Árbol (d)



Árbol (f)



Árbol (h)



Árbol (i)

El árbol (h) corresponde al programa que se necesita para el cálculo de la conversión de grados.

La operación cruce crea nuevos hijos intercambiando subárboles entre los dos padres. Como se intercambian subárboles completos, la operación cruce siempre produce programas sintáctica y semánticamente válidos, como descendientes sin considerar el punto de cruce; además son ejecutables en un computador, por lo cual la medición de su aptitud es directa.

Este nuevo enfoque enriquece enormemente el campo de la

aptitud dio que seleccionamos dos copias de los árboles b, d, f. Esta es entonces la población de la segunda generación.

programación y diseño de algoritmos, y es un campo promisorio como un nuevo método para la solución de problemas en ingeniería. Por supuesto que hasta ahora es una materia incipiente y por esta razón, es un camino que todavía tiene mucho terreno sin explorar, y que promete un gran espectro de aplicaciones. En la Universidad se han tenido experiencias en el campo del Hardware Evolutivo, para la obtención de circuitos combinatoriales con base en tablas de verdad y en un modelo predador - presa, en el cual se evolucionan los comportamientos del predador y la presa.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Koza, J. R.**, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press: Cambridge MA, 1992.
2. **Martínez J., Rojas S.**, Informática Evolutiva, Facultad de Ingeniería, Universidad Nacional, Bogotá 1999.