

# CC40A Auxiliar

## Algoritmos Aleatorios

Profesor: Ricardo Baeza Yates  
Prof. Auxiliar: Rodrigo Paredes

Dept. de Ciencias de la Computación, Universidad de Chile.

### 1. Preliminares

Los algoritmos genéticos son unos de los métodos basados en la biología, al igual que las redes neuronales. Esta técnica pretende capturar los mecanismos de adaptación de las especies al ambiente en que viven, favoreciendo a la evolución y la supervivencia de los individuos mejor adaptados permitiendo su reproducción y el descarte de los más débiles. Para esto se consideran los siguientes elementos:

**Población:** subconjunto de soluciones factibles del problema que se están evaluando actualmente. Sea  $P(t) = \{x_1^t, \dots, x_n^t\}$  la población en el instante  $t$ . La población representa al conjunto de individuos sobrevivientes de la especie.

**Gen:** un individuo o elemento de la población. Cada elemento  $x_i^t$  representa una solución potencial del problema. El gen debe ser diseñado de modo de modelar la estructura de la solución del problema y ser capaz de resolverlo. También se conocen como *cromosoma* o *genotipo*.

**Fitness:** es la función que mide la adaptación del individuo. Se utiliza para evaluar la calidad del gen.

**Operadores genéticos:** son los operadores que permiten reproducir la población a partir de la población actual.

En términos muy simplificados, los algoritmos genéticos intentan resolver problemas de optimización mediante un acercamiento iterativo hacia la solución del problema. Para esto, el proceso comienza con una población inicial escogida de modo uniforme, de modo de explorar simultáneamente múltiples zonas del universo de soluciones factibles del problema, es decir, evitar el sesgo de la solución. Para iteraciones posteriores, se produce una nueva población seleccionando los individuos mejor adaptados, y reproduciéndolos mediante los operadores genéticos. Idealmente, el proceso termina cuando la solución se estabiliza, es decir, no hay mejoras en la función objetivo, en la práctica esto no siempre se consigue, luego también se suele detener el proceso luego de un número fijo de iteraciones. El esquema básico de un algoritmo genético se muestra en la Figura 1.

Los operadores genéticos se utilizan para reproducir nuevas generaciones a partir de los sobrevivientes de la generación actual (dado  $A(t)$  se reproduce  $P(t + 1)$ ). Los operadores genéticos más comunes son el *crossover* (*cruce* o *recombinación genética*) y el operador de *mutación*. El operador de crossover consiste en que dado dos genes se produce otro par de genes que comparten material genético, teniendo el cuidado de que los genes producidos

```

Inicializar  $P(0)$ 
while (condTermino = FALSE)
  evaluar  $P(t)$  con la función de fitness
   $A(t) \leftarrow$  selecciona individuos de  $P(t)$ 
   $P(t+1) \leftarrow$  reproduce la población a partir de  $A(t)$ 
  t++

```

Figura 1. Esquema de un algoritmo genético.

resparenten la estructura del gen (se corresponde a la reproducción sexuada). El operador de mutación consiste en modificar una componente de gen al azar (se corresponde a la reproducción asexual). Estos operadores se muestran en la Figura 2.

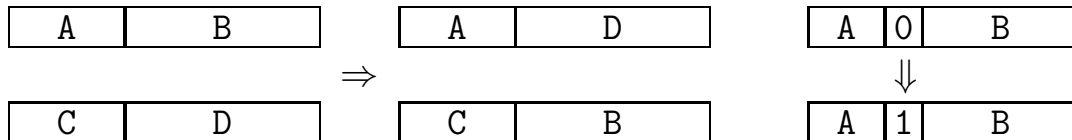


Figura 2. Operadores genéticos usuales. A la izquierda, crossover de dos genes. A la derecha, mutación de un bit escogido al azar.

Inicialmente los genes se representaron con secuencias binarias de largo fijo, lo que permitía una aplicación sencilla de los operadores genéticos comunes. Debido a que esta representación es poco flexible, se liberó la estructura de secuencia binaria adaptándola a otra que sea apropiada para la aplicación, con lo cual, también es necesario adaptar los operadores genéticos.

Por último, es necesario indicar que en aplicaciones prácticas, los algoritmos genéticos se utilizan cuando no se conocen soluciones eficientes para el problema en estudio.

## 2. Calcular la raíz cuadrada de $n$

Este es un ejemplo clásico de la aplicación de algoritmos genéticos, que consiste en calcular el entero más cercano a la raíz cuadrada de  $n$ .

**gen:** números enteros representados como secuencias de bits (largo acotado por la representación binaria de  $n$ ).

**poblacion inicial:** secuencias de bits (distribuidas uniformemente), acotadas entre  $0 \dots n$ .

**fitness:**  $\min |n - gen^2|$ .

**operadores geneticos:**

- cruce: intercambiar bits.
- mutacion: cambiar un bit al azar.

**condicion de corte:** por timeout (ej 100 generaciones) o por error pequeño.

**esquema global:** el de siempre.

### 3. Construir el MST de un grafo

Este es un ejemplo sólo tiene fines didácticos, puesto que existen algoritmos eficientes para calcular el MST (Kruskal y Prim, entre otros).

**gen:** colección de arcos.

**poblacion inicial:** permutaciones  $1 \dots N$ .

**fitness:**  $\min \sum_{e \in MST} w(e)$ .

**operadores geneticos:**

- Mutacion adaptada a MST: tomar al azar un arco, meterlo al MST y sacar el mas caro.
- Cruce adaptado a MST: buscar en los dos MST un grupo de nodos que esten conectados entre si e intercambiar los arcos

**condicion de corte:** por timeout (ej 100 generaciones) o por error pequeño.

**esquema global:** el de siempre.