

# Facultad de Ciencias UNAM

## Programa 03

*Profesora: María de Luz Gasca Soto*  
*Ayudante: José Luis Vázquez Lázaro*

24 de noviembre de 2018

**Fecha de entrega:** martes 27 de noviembre antes de las 23:59 hrs.

### Objetivos.

1. Implementar un algoritmo genético para el Problema MAX-SAT.
2. Aplicar la metaheurística a un ejemplar con características específicas.

### Resumen.

Los algoritmos genéticos:

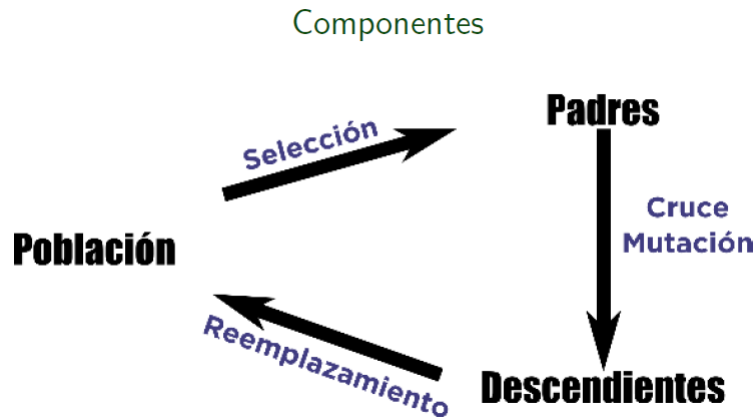
- Se inspiran en la evolución biológica.
- Hacen evolucionar poblaciones.
- Modifica aleatoriamente las características de los individuos.
- Hacen una selección de los mejores individuos.
- Poseen un modelo iterativo.

Los componentes esenciales de un algoritmo genético son:

- Población (conjunto de individuos que son soluciones factibles del problema).
- Proceso de selección.

- Proceso de reproducción (cruce/mutación).
- Proceso de reemplazo.

El principio de funcionamiento de los algoritmos genéticos se basa en someter a una población de individuos a una serie de transformaciones (proceso de reproducción), para hacer una búsqueda, así como a un proceso de selección, que favorece a los mejores individuos de la misma. De esta manera, cada uno de los ciclos de selección-reproducción, sufridos por la población, constituyen una generación; se espera que al cabo de un determinado número de generaciones, la población halla evolucionado hacia la solución óptima al problema, o que al menos se esté muy próximo a ella. Todo este ciclo de creación de una nueva generación puede verse claramente y de forma simplificada en la siguiente figura.



## Desarrollo.

### 1. Codificación.

- El ejemplar  $I = (\{x_1, x_2, \dots, x_n\}, \{c_1, c_2, \dots, c_m\})$  que reciba tu heurística deberá estar codificado a través de una matriz  $F$  de tamaño  $m \times n$ , donde
  - \* Para cada  $i \in \{1, 2, \dots, m\}$ , el renglón  $i$  corresponderá a la cláusula  $c_i$ .
  - \* Para cada  $j \in \{1, 2, \dots, n\}$ , la columna  $j$  corresponderá a la variable  $x_j$ .

- \* La entrada  $F_{ij}$  indicará si la cláusula  $c_i$  contiene o no a la variable  $x_j$ , para toda  $i \in \{1, 2, \dots, m\}$  y  $j \in \{1, 2, \dots, n\}$ .
- \* Deberás considerar las siguientes restricciones:  $100 \leq n$ ,  $50 \leq m \leq 60$  y  $3 \leq |c_i| \leq 5$  para toda  $i \in \{1, 2, \dots, m\}$ .
- Cada individuo de la población será una *asignación de verdad* para  $\{x_1, x_2, \dots, x_n\}$  (es decir, una solución factible para  $I$ ) y se codificará como un vector (arreglo) de bits de tamaño  $n$ , donde la entrada  $j$  del arreglo será el valor de verdad de la variable  $x_j$ , para toda  $j \in \{1, 2, \dots, n\}$ .

## 2. Población inicial.

- Deberás crear una población inicial de manera conveniente.

## 3. Función de evaluación (*fitness*).

- La “aptitud” de cada individuo estará determinada por el número total de cláusulas que satisface la *asignación de verdad* que codifica.

## 4. Proceso de selección.

- Deberás definir cómo seleccionar a los individuos que serán padres, utilizando la función de evaluación.
- Debes considerar que, los individuos más aptos deben tener mayor probabilidad de ser elegidos y los individuos menos aptos deben tener oportunidad de ser elegidos.

## 5. Proceso de reproducción.

- **Cruce.** Deberás implementar los siguientes operadores de cruce: *Partially Mapped Crossover* y *Order Crossover*. En cada iteración de tu heurística, esta deberá elegir (de manera aleatoria) uno de estos tres operadores de cruce para aplicarlo.
- **Mutación.** Deberás implementar los siguientes operadores de mutación: *Displacement Mutation*, *Exchange Mutation*. En cada iteración de tu heurística, esta deberá elegir (de manera aleatoria) uno de estos dos operadores de mutación. La probabilidad de aplicar cualquiera de estos dos operadores de mutación deberá ser de a los más 0.2.

## 6. Proceso de reemplazo.

- Deberás definir cómo reemplazar a los individuos de la población, por aquellos obtenidos en el proceso de reproducción.
- Considera que, es posible decidir no reemplazar al mejor individuo de la población.

#### 7. Terminación.

- Deberás definir una condición de parada adecuada. Por ejemplo, fijar un número máximo de iteraciones o terminar si después de un cierto número de iteraciones no hay mejora.

### Aplicación.

Deberás aplicar tu algoritmo genético a un ejemplar de MAX-SAT generado de manera aleatoria bajo la codificación y restricciones indicadas en el primer punto de la sección anterior.

### Indicaciones.

1. Tu programa deberá mostrar en pantalla su entrada y salida de manera adecuada para el usuario.
2. Se recomienda utilizar un lenguaje de programación orientado a objetos.

### Formato de entrega.

Deberás enviar un archivo .zip, con nombre <ApP><ApM><Nombre>-P03, que contenga lo siguiente:

- La implementación solicitada.
- Un archivo README con las instrucciones para compilar y ejecutar tu programa.

a la dirección de correo `luis_lazaro@ciencias.unam.mx` con el asunto `[CC20191]Programa03`.