

Práctica 5

David Gabriel Palmerin Morales

April 2, 2019

1. **Tenemos dos cromosomas padres distintos en un SGA con codificación binaria, cada uno con L bits. Seleccionamos aleatoriamente un punto de cruce $c \in \{1, \dots, L-1\}$ y realizamos la cruce de un punto. ¿Cuál es la probabilidad de que ambas soluciones hijas sean clones (idénticas) de sus padres?**

Sean h_1 y h_2 los hijos de P_1 y P_2 . Donde los hijos son obtenidos a partir del operador de recombinación con P_1 y P_2 en un punto c .

Supongamos sin pérdida de generalidad que los hijos son de la siguiente forma, donde \parallel denota concatenación:

$$\begin{aligned}h_1 &= P_1[0 : c] \parallel P_2[c : L] \\h_2 &= P_2[0 : c] \parallel P_1[c : L]\end{aligned}$$

Buscamos que los hijos sean idénticos a los padres, por lo que tendremos los dos casos siguientes:

- Si $P_1[0 : c] == P_2[0 : c]$
Cómo $P_1[0 : c] == P_2[0 : c]$, entonces $h_1 == P_2$ y $h_2 == P_1$. De esta forma tenemos que ambos hijos son iguales a los padres.
Dicho esto, basta calcular la probabilidad de que $P_1[0 : c] == P_2[0 : c]$, dado que tenemos 2^L genomas en total, entonces tendremos lo siguiente:

$$\frac{2^L}{2^L} \cdot \frac{2^{L-c}}{2^L} = 1 \cdot \frac{1}{2^c} = \frac{1}{2^c} \quad (1)$$

Esto resulta pues el primer padre puede seleccionar cualquier genoma de todos los disponibles, es decir $\frac{2^L}{2^L}$, sin embargo, el segundo padre deberá seleccionar un genoma que tenga el mismo contenido en $[0 : c]$, de esta forma tenemos $\frac{2^{L-c}}{2^L}$ pues se fijan los primeros c genes.

- Si $P_1[c : L] == P_2[c : L]$
El primer padre podrá seleccionar un genoma de todos los disponibles, es decir, tendrá $\frac{2^L}{2^L}$.
Por otro lado, el otro padre deberá cumplir la suposición inicial

$P_1[c : L] == P_2[c : L]$, por lo que tendrá $\frac{2^c}{2^L}$ pues únicamente fijará $L - c$ genes. De esta forma obtendremos:

$$\frac{2^L}{2^L} \cdot \frac{2^c}{2^L} = 1 \cdot \frac{2^c}{2^L} = \frac{2^c}{2^L} \quad (2)$$

Finalmente, dado que basta con que ocurra uno de los dos incisos anteriores, entonces tenemos que son eventos independientes por lo que tenemos la siguiente probabilidad:

$$P = \frac{1}{2^c} + \frac{2^c}{2^L}$$

2. El problema one-max consiste en buscar una cadena binaria de L bits con el mayor número de unos posible. Es decir, el fitness de una cadena es el número de unos que contiene, por lo que se quiere maximizar $f(x) = \sum_{i=1}^L x_i$ con $x_i \in \{0, 1\}$. Por supuesto que podemos resolver este problema de forma sencilla escribiendo 1s consecutivamente en la solución, pero estamos interesados en ver si el SGA puede resolverlo. Implementa en Python el SGA usando:

- Selección proporcional
- Cruza de un punto con $P_c = 0.7$
- Mutación bit a bit con $P_m = 0.01$
- Tamaño de cromosoma $L = 30$
- Limite de generaciones: 100
- Tamaño de población: 20

Ver archivo *Practica5.py*

Uso: `python3 Practica5.py 2`

3. Usando el paquete SCHEMATAX de Python: <https://github.com/iSTB/python-schemata> vamos a validar de forma experimental el teorema de los esquemas de J. Holland, calculando los esquemas procesados en cada iteración del SGA para observar los bloques contructores.

Ver archivo *Practica5.py*

Uso: `python3 Practica5.py 3`