Examen I

Rodrigo Hernández Mota (if 693056)

September 15, 2016

0.1 I

El algorítmo genético permite optimizar una función de la forma $F(X)$, en donde $X = x_1, ..., x_n$ con un método heurístico que pretende simular el sistema de recombinación genético biológico.

0.1.1 Inicialización

Dada una región de busqueda se inicializa la población de 'm' pobladores. Los pobladores son distintos valores X para los cuales la función puede ser evaluada.

0.1.2 Selección

Se evalúa el desempeo de cada poblador mediante el valor resultante de la funci'on F. Dependiendo si el objetivo es maximizar o minimizar, se seleccionan los pobladores más aptos. Se deberán escoger al menos dos para asegurar la recombinación genética.

0.1.3 Cruzamiento

Habiendo seleccionado los mejores pobladores, se utiliza su código genético para generar nuevos pobladores (nueva generación). En esta generación estan los mejores pobladores anteriores y los nuevos resultantes del cruzamiento.

Existen diferentes métodos de cruzamiento.

0.1.4 Mutación

Para evitar la degeneración de la población se establece un factor de mutación. Parte de la nueva generación sufre alteraciones en su código genético (generalmente representado como un numero en binario).

Los pasos a partir de la inicialización son repetidos de forma iterativa. De esta forma la población mejora (o intenta mejorar) su desempeo generación tras generación.

0.2 II

Sea F una función de x_1, x_2 en donde $minF(x_1, x_2)$ se encuentra en los intervaloes $1 \leq x_1 2$ y $-2 \leq x_2 \leq 2$, entonces si se requiere que el algorítmo pueda buscar en intervalos de $2.44x10^{-4}$, el número de bits necesarios son: 14.

Declaramos Δ como una variable para determinar el "salto" o cambio que se dará debido a la división o discretización (determinado por los bits) del espacio de búsqueda real.

$$\Delta = \frac{ls - li}{2^n - 1}$$

En donde ls, li son los límites superior e inferior y n es el numero de bits. Entonces calculamos delta para cada variable:

$$\Delta_1 = \frac{1}{m}$$

$$\Delta_2 = \frac{4}{m}$$

Debido a que $1/m < 4/m$ entonces para asegurar que el algorítmo pueda buscar en (al menos) en el intervalo deseado; se calcula m de la siguente forma:

$$m = 4 * \Delta_2^{-1} = \frac{4}{2.44x10^{-4}} = 16393.4426$$

Despejando n y sustituyendo m en $2^n - 1 = m$ tenemos:

$$n = \log_2 m + 1 = \log_2 16394.4426 = 14.000919$$

Debido a que el "numero de bits" en práctica debe ser entero; se rendondea al enterno ms cercano;

$$n = 14$$

0.3 III

Sea la F la función por partes:

$$F(x_1, x_2) = \begin{cases} x_1^2 + x_2^2 & : x_1 + x_2 \geq 1 \\ -10(x_1 + x_2) + 20 & : x_1 + x_2 < 1 \end{cases}$$

Considerando la reestriccón de $x_1, x_2 > 1$ entonces el valor de X que minimiza F es:

$$x_1 = 0.4983$$
$$x_2 = 0.5101$$

Generando un $F = 0.500$.

0.4 IV

Utilizando la descripcin y datos del problema, se propone una funcin:

$$F = f(x_1, x_2, x_3, x_4, x_5)$$

Sujeto a las reestricciones:

$$\begin{gathered} {}_1 \geq 48 \\ x_1 + x_2 \geq 79 \\ x_1 + x_2 \geq 65 \\ x_1 + x_2 + x_3 \geq 87 \\ x_2 + x_3 \geq 64 \\ x_3 + x_4 \geq 73 \\ x_3 + x_4 \geq 82 \\ x_4 \geq 43 \\ x_4 + x_5 \geq 52 \\ x_5 \geq 15 \end{gathered}$$

En donde F representa el costo de la empresa (pago a empleados), $x1, \ldots, x5$ son el numero de empleados por turno $i = 1, 2, ...$ y las reestricciones son el numero mínimo de agentes necesarios.

Resultado (número de empleados por turno):

$$
\begin{aligned}
{}_1 &= 50 \\
x_2 &= 29 \\
x_3 &= 39 \\
x_4 &= 43 \\
x_5 &= 16
\end{aligned}
$$

Generando un costo (aprox.) de 30676.5198449 unidades monetarias.