



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL IPN

RMHC

TECNOLOGÍAS DE LA PROGRAMACIÓN

Miembros del equipo:
Hernán Guillermo Dulcey Morán
Karla Jacquelin Guzmán Sánchez

19 de Noviembre de 2019

Introducción

El algoritmo genético RMHC (Random Mutation Hill Climbing) es un algoritmo que está diseñado para cambiar los "genes" de un "individuo" aleatoriamente con el fin de obtener una mejor versión que se acomode a la necesidad del problema.

Se considera un individuo como un conjunto de características que van a cambiar dependiendo de las instrucciones que reciba. En este caso el individuo va a cambiar un arreglo propio que se denomina "genes", que están constituido de bits (cero o unos).

Cuando el individuo muta, cambia el valor de una de sus posiciones por otro valor. Es decir, si el valor es uno pasa a ser cero, y si era cero pasa a ser uno.

Problema

El problema que se desea resolver con RMHC es un problema de maximización. El algoritmo va a evaluar una función y va a tratar de encontrar el mayor valor para un rango de valores determinado. RMHC hará que el individuo mute un determinado número de veces e irá evaluando cada una de esas mutaciones conservando las mejores. Al terminar presentará la mejor versión de su individuo. Cabe resaltar que en muchos casos no es posible llegar a la mejor solución. Esto dependerá de el número de mutaciones realizadas y que el individuo no caiga en un optimo local.

Representación

El valor de x vendrá representado por un vector de ceros y unos, el cual en su conjunto representan un número real en binario. Cada gen tiene un equivalente a una potencia de dos:

$$[1, 0, 1, \dots, 1, 0, 1] = 2^n * (1) + 2^{n-1} * (0) + 2^{n-2} * (1) + \dots + 2^2 * (1) + 2^1 * (0) + 2^0 * (1)$$

Por ejemplo:

$$[1, 0, 1] = 2^2 * (1) + 2^1 * (0) + 2^0 * (1) = 4 + 0 + 1 = 5$$

De igual forma, la parte decimal puede ser representada de la siguiente manera:

$$[1, 0, 1, \dots] = 2^{-1} * (1) + 2^{-2} * (0) + 2^{-3} * (1) + \dots$$

Por ejemplo:

$$[1, 0, 1] = 2^{-1} * (1) + 2^{-2} * (0) + 2^{-3} * (1) = 0.5 + 0.25 + 0.125 = 0.875$$

Es posible combinar las dos partes para representar un número real, para ello es necesario delimitar hasta donde va la parte entera y desde donde empieza la decimal.

Problemas de representación

Un problema de representación consiste en que dado un valor, el algoritmo no puede mutar al siguiente valor sin la posibilidad de cambiar dos genes al tiempo.

Por ejemplo se tienen 3 genes para buscar maximizar la función $f(x) = x$, donde el rango está entre 0 y 6:

$$rango = 0 \leq x \leq 6$$

Si el individuo llega al valor de 5:

$$[1, 0, 1] = 5$$

Le es imposible cambiar a la mejor solución, puesto que prender el gen faltante implica llegar a una respuesta fuera del rango. Así mismo apagar cualquier gen implica llegar a una peor respuesta.

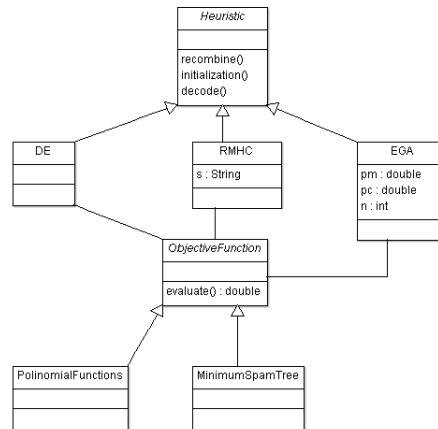
Implementación

El algoritmo RMHC se implementó en Python, con las siguientes características.

Como componente principal se tienen dos clases abstractas: *Heuristic* y *ObjectiveFunction*.

La primera será la clase padre para el algoritmo RMHC y algunos otros si se desea en un futuro. La segunda es la clase padre de las funciones a evaluar, siendo sus hijos funciones específicas a evaluar.

Para implementarlo se tomó como base el siguiente diagrama de clases:



Conclusión

Este algoritmo tiene sus limitantes, pero como para todo algoritmo voráz, sus respuestas se acercan mucho a la mejor, dependiendo de la cantidad de tiempo que se le de para trabajar. Una de sus mayores limitantes es el problema de entrar en óptimos locales y no tener una clara forma de salir de ellos.