Universidad Simón Bolívar

Departamento de Computación y T.I

Inteligencia Artificial II

CI-5438

Septiembre-Diciembre 2014

**INFORME PROYECTO 2 – GABIL**

**Profesora: Integrantes:**

Ivette Carolina Martínez Carla Barazarte 08-10096

Ramón Márquez 10-10849

Grupo 10

Sartenejas, diciembre de 2014

**INFORME**

**Resumen:**

El presente proyecto se encarga de crear un clasificador, utilizando un algoritmo genético, que permita clasificar las tres clases de la flor Iris (Setosa, Versicolor, Virginica). El algoritmo genético empleado está basado en el sistema GABIL[1]. Además, se utiliza un librería de python para facilitar la implementación del algoritmo; este se llama “pyevolve”.

En este experimento se requiere realizar distintas codificaciones para la selección de los padres y de los sobrevivientes. Sin embargo, una de las funciones a utilizar es el de “rueda de ruleta”. De igual manera, se realizará un mecanismo de penalización de los tamaños de los clasificadores en la función fitness y compararlo con los resultados de no tener dicha penalización.

**Descripción de la implementación:**

Inicialmente, se utiliza una escala fijada para transformar a cada individuo en valores 0's y 1's. Para la implementación se utiliza una librería de python llamada “pyevolve” el cual permite codificar algoritmos genéticos. Cabe mencionar que esta librería no te permite modificar el tipo de selección de los padres pero si lo permite con los sobrevivientes. Por otra parte, la librería permite crear nuestros propios operadores de crossover y del fitness el cual nos es útil puesto que lo adaptamos de acuerdo al sistema GABIL. Para la operación de mutación, pyevolve te da las herramientas para usarlo. En nuestra implementación, usamos el método swap para la mutación.

El otro método empleado para selección de los sobrevivientes, además del de “rueda de ruleta”, es el “rank selector”. Este selector escoge el mejor individuo de la población cada vez. En el caso de la penalización, lo que se hace es dividir el fitness que se obtiene entre el tamaño del individuo.

Para dar un resultado, se realizan 10 corridas con cada configuración debido a que los algoritmos genéticos son estocásticos.

**Descripción del algoritmo genético:**

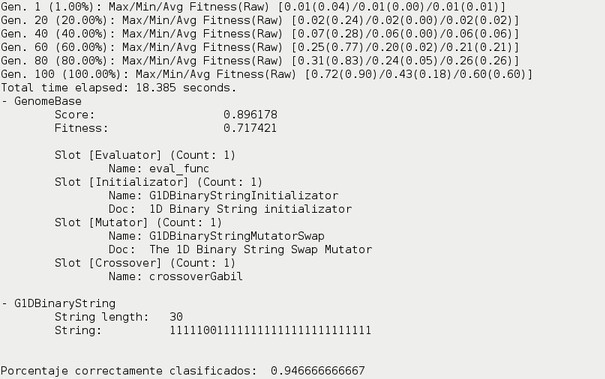
Como se dijo anteriormente, el algoritmo genético empleado está basado en el sistema GABIL. Los parámetros que se usaron para la tasa de mutación son: 0.001, 0.015 y 0,02. Para la tasa de crossover se usaron los siguientes valores: 0.6, 0.7 y 0.85.

**Descripción y análisis de los experimentos realizados:**

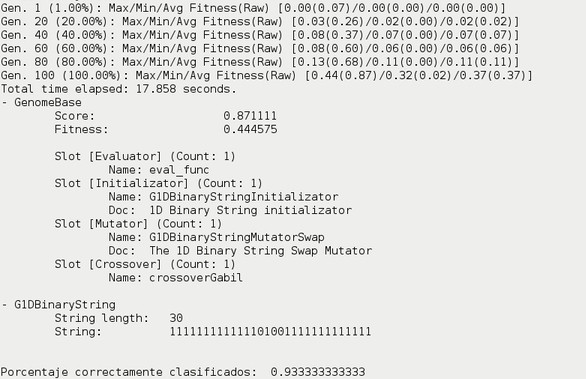
**Selección por “rueda de ruleta” y sin penalización**

**(Tasa de mutación, Tasa de Crossover)**

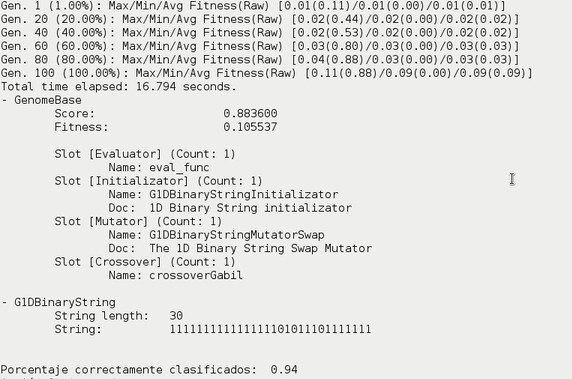
**(0.001 , 0.6)**

****

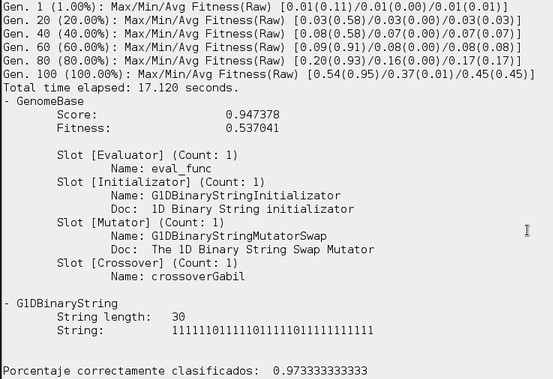
**(0.015 , 0.6)**

****

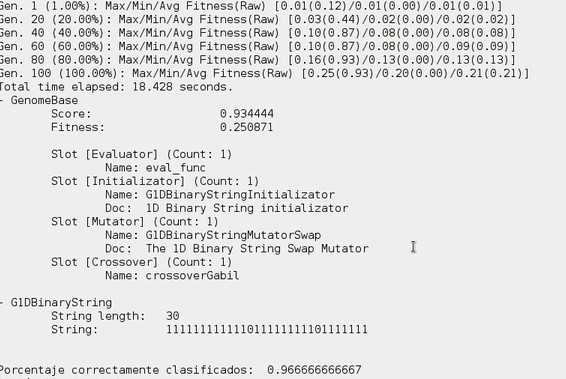
**(0.02 , 0.6)**

****

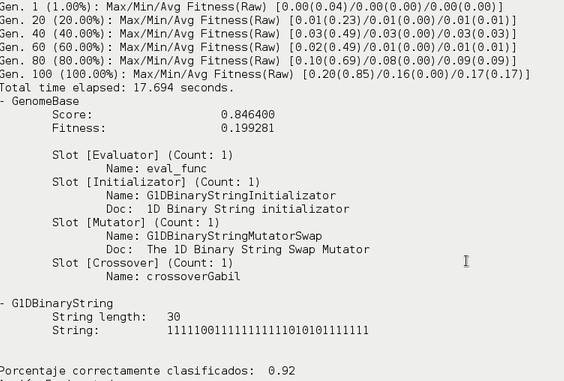
**(0.001 , 0.7)**

****

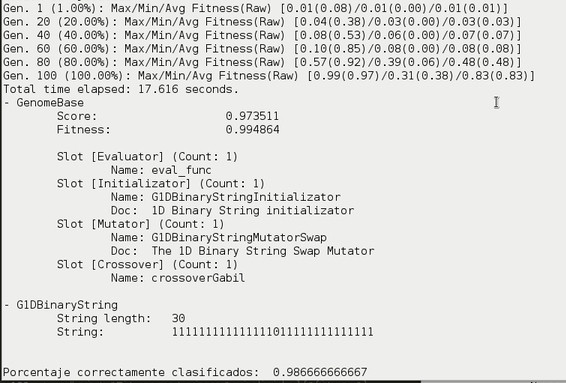
**(0.015 , 0.7)**

****

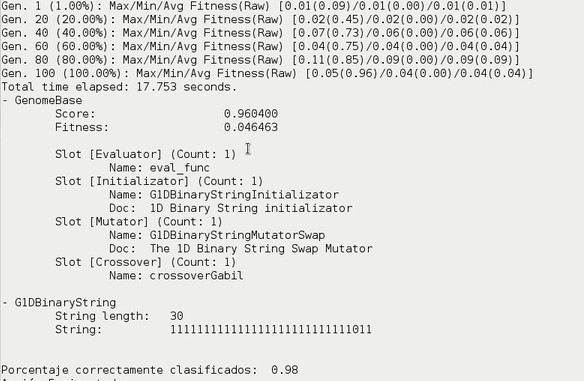
**(0.02 , 0.7)**

****

**(0.001 , 0.85)**

**FIGURA 1**

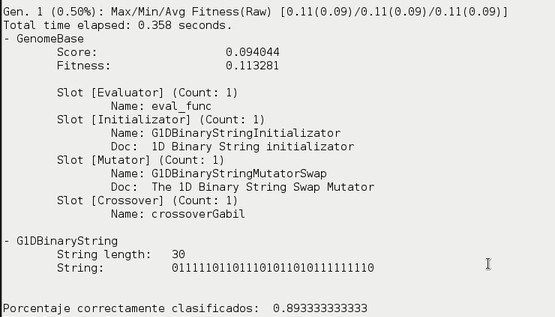
**(0.015 , 0.85)**

****

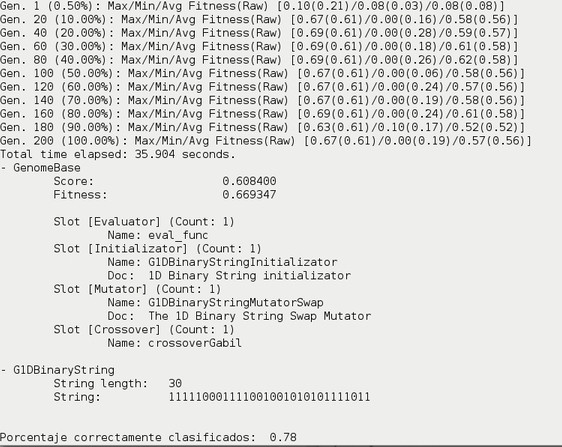
Podemos notar que esta configuración da buenos resultados. El porcentaje de clasificados por el mejor individuo obtenido en todos los casos da mayor al 90% y muy parecidos en todos los casos.

A continuación se darán algunos resultados por el otro método de selección “**rank selector” sin penalización**:

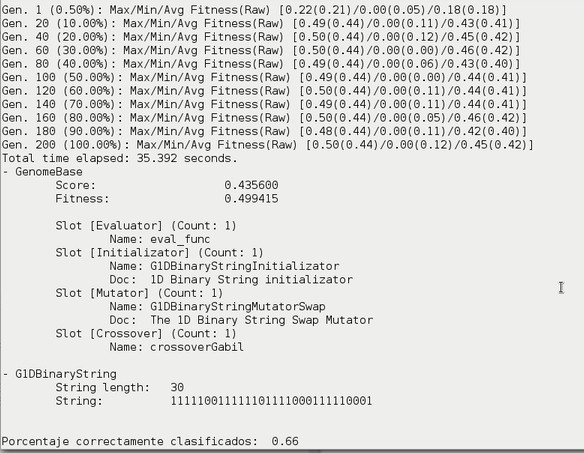
**(0.001 , 0.6)**

****

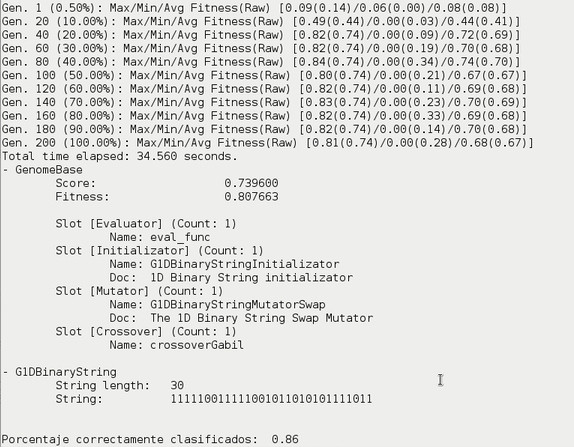
**(0.015 , 0.6)**

****

**(0.015 , 0.7)**

****

**(0.015 , 0.85)**

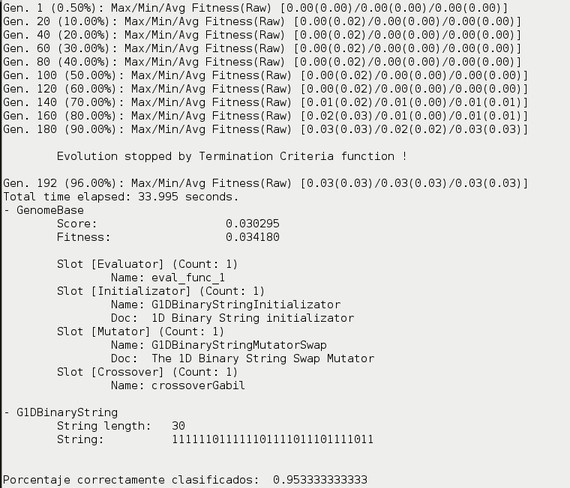
****

Podemos notar que con este tipo de selección los resultados son relativamente buenos, pero el mejor sigue siendo el método “rueda de ruleta”.

A continuación, se presentará el mejor resultado obtenido utilizando la penalización en la mejor configuración:

**Selección por “rueda de ruleta” y con penalización**

**(0.001 , 0.6)**

****

Con los resultados obtenidos tenemos que la mejor configuración obtenida es la siguiente: tasa de mutación de 0.001, tasa de crossover de 0.85 y utilizando el método de selección “rueda de ruleta” el cual arroja un porcentaje de ejemplos bien clasificados de aproximadamente 98.667%. De igual manera, el mejor conjunto de reglas hallado está presente en “FIGURA 1”.

La función fitness empleada es la descrita en el sistema GABIL. Este toma un individuo y lo compara con cada ejemplo de la población dada en el archivo de entrada y que ha sido codificado en 0's y 1's. Cada ejemplo es comparado con cada regla del individuo. Si una de sus reglas hace match con el ejemplo entonces le suma 1 a una variable temporal que registra la cantidad de ejemplos con las que el individuo hace match. Luego, se divide la suma total entre el tamaño de la población y el resultado que de se eleva al cuadrado y es lo que se retorna. En cuanto a la penalización, ayuda a reducir el tiempo de ejecución del algoritmo y no es muy necesario puesto que para el problema no hace falta tener un número de reglas grandes.

**Referencias:**

[1] Tom M. Mitchell, Machine Learning, McGraw-Hill, 1997..