

Centro Universitario de Ciencias Exactas e Ingenierías

Licenciatura en Ingeniería en Computación

Materia: Seminario de Solución de Problemas de Inteligencia Artificial I. Clave: I7039.

Profesor: Sencion Echauri Felipe

Estudiante: Silva Moya José Alejandro. Código: 213546894.

**Actividad 9: Differential Evolution**

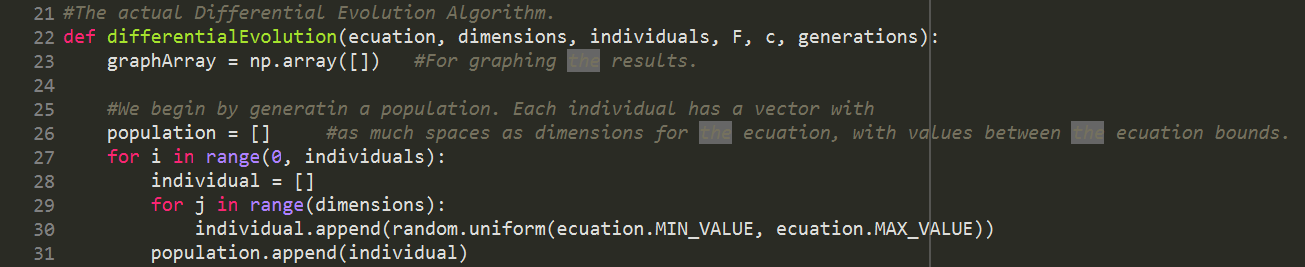
**Instrucciones:** Implementar y evaluar el rendimiento del algoritmo de optimización por evolución diferencial (differential evolution) para las siguientes funciones:

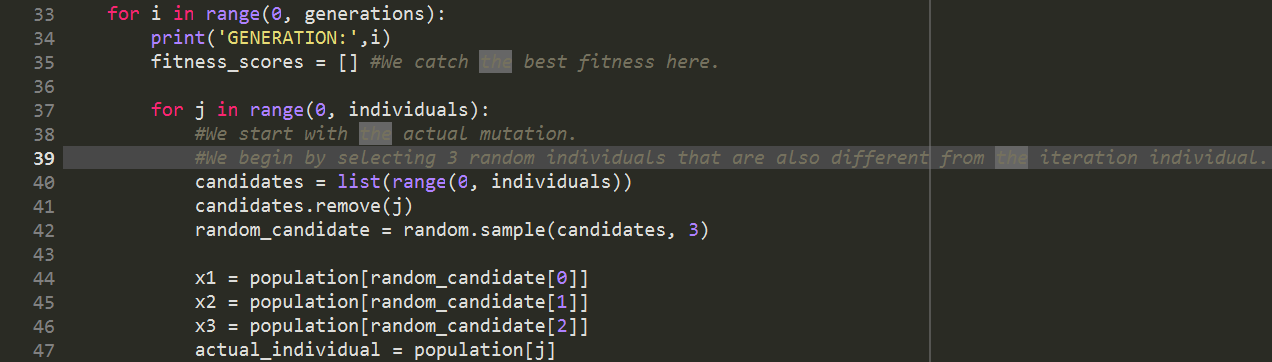
* Sphere
* Rosenbrock
* Rastrigin
* Quartic

Para cada función realizar 5 ejecuciones con 2, 4, 8 y 16 dimensiones, cada ejecución se detendrá a las 2000 generaciones.

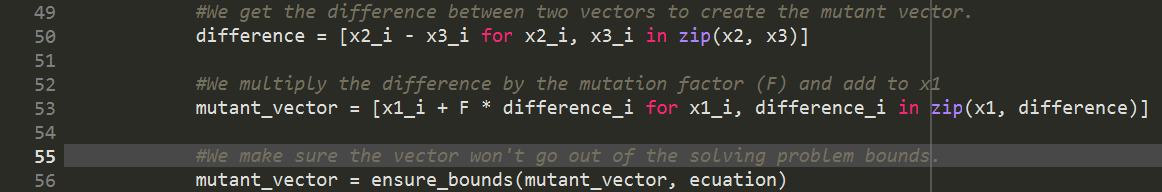
Se deberá graficar el comportamiento del algoritmo; para ello se deberá promediar el valor del mejor fitness de las 5 ejecuciones en la generación 0, 100, 200, ... 2000. Se deberá generar una gráfica para cada dimensión y además una gráfica en la que se incluyan las ejecuciones para 2, 4, 8 y 16 dimensiones, es decir un total de 5 gráficas por función.

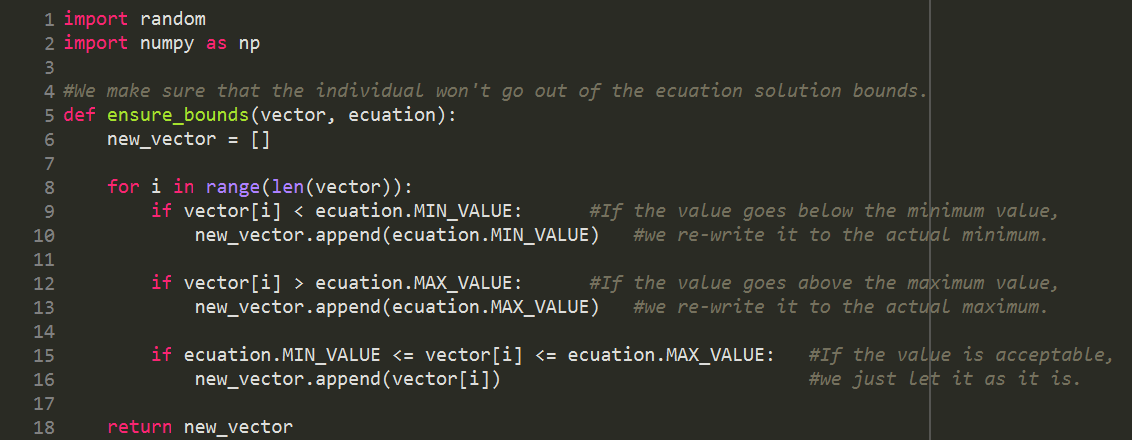
**Desarrollo**

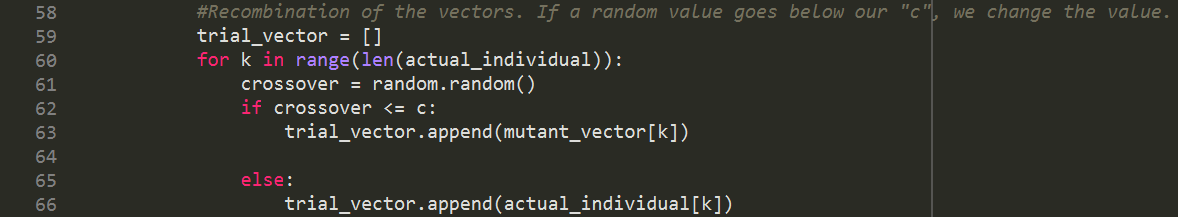
Comenzamos por generar una población de individuos, que tendrán un vector de tamaño igual a la cantidad de dimensiones del problema, con valores aleatorios entre los límites de búsqueda de solución del problema.

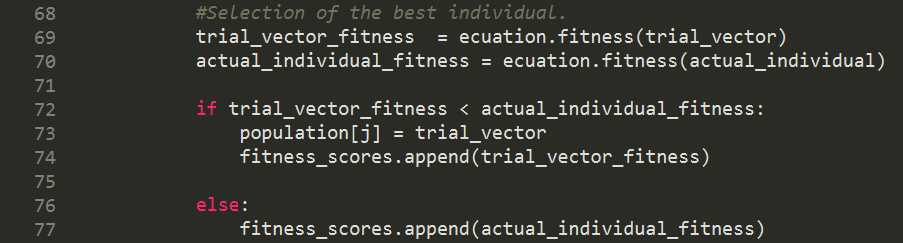
Ahora procedemos para cada individuo existente de la población:

* Tomamos 3 individuos aleatorios que sean únicos entre sí y también diferentes al individuo correspondiente a la iteración. Esto lo hacemos para poder crear el vector nuevo mutado.

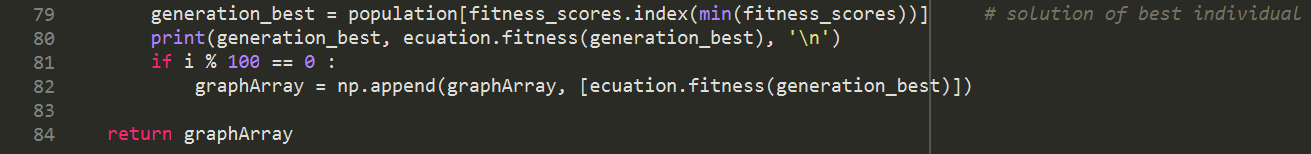
En cada iteración obtenemos las diferencias por posición del vector 2 con respecto del 3, y luego generamos el nuevo vector mutado, sumando a cada posición del vector tomado en el punto anterior y sumándole nuestro valor F multiplicado por las diferencias obtenidas también en el paso anterior.

Dado que necesitamos revisar que las soluciones no se salgan del espacio de solución de ecuación, si encontramos que se salen en positivo o negativo, los regresamos a los límites, y en cualquier otro caso los dejamos igual, ya que serían aceptables.

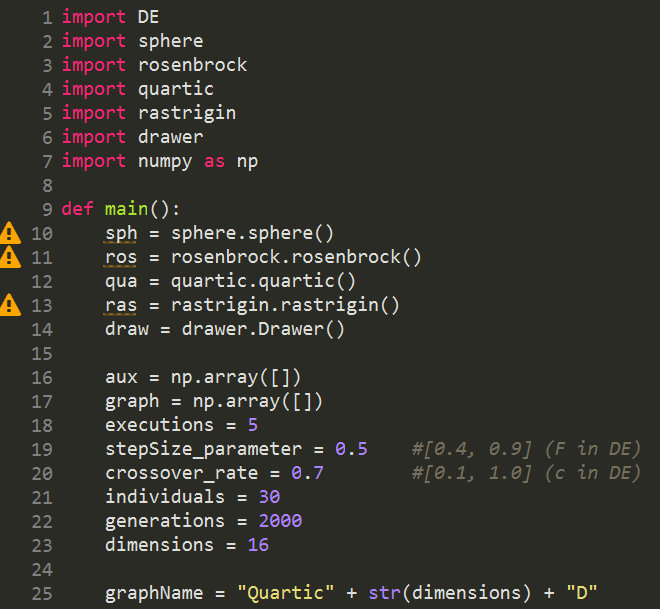
Iniciamos con la recombinación. Generamos un número aleatorio correspondiente al rango de dimensiones, un vector donde haremos la recombinación, y con un número aleatorio entre 0.1 y 0.9, si se cumple que sea menor que nuestro valor aleatorio c, intercambiamos el valor del vector original con el del vector mutado; en caso contrario, lo dejamos como está.



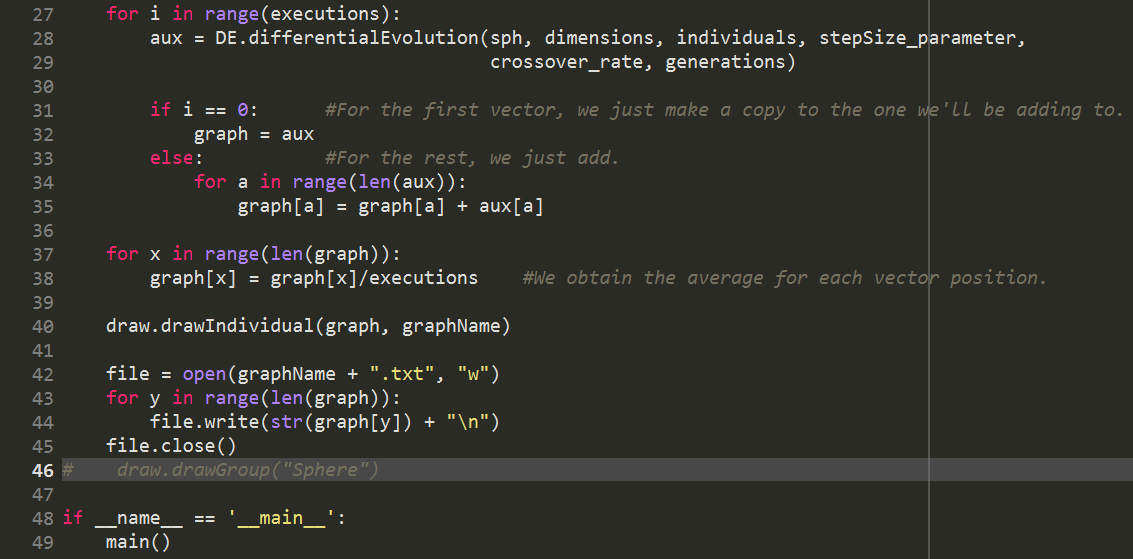
Finalmente evaluamos el fitness de nuestros resultados. Si el fitness del vector resultante es mejor que el actual, reemplazamos, y en caso contrario, lo dejamos como está.



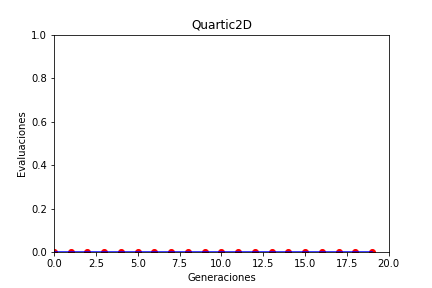
Imprimimos el mejor resultado de cada generación, y cada 100 generaciones obtenemos el mejor fitness para poder graficarlo posteriormente.

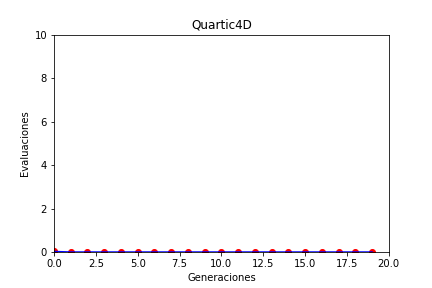


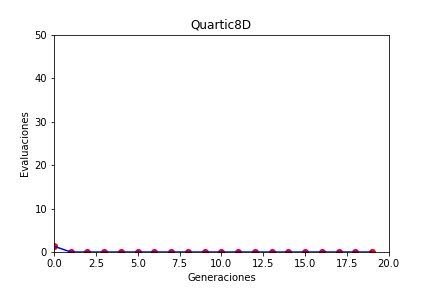
En el main tenemos objetos de cada ecuación a resolver, un objeto para graficar individual y en grupo, y los parámetros específicos que necesita el algoritmo para funcionar.

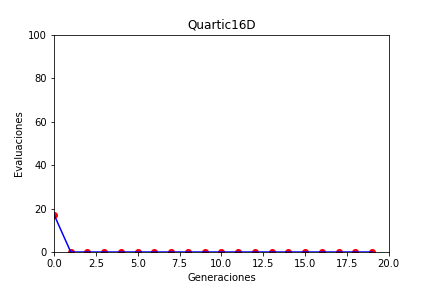
El resto del algoritmo (así como el fitness de las ecuaciones), son exactamente las mismas que en las entregas anteriores.

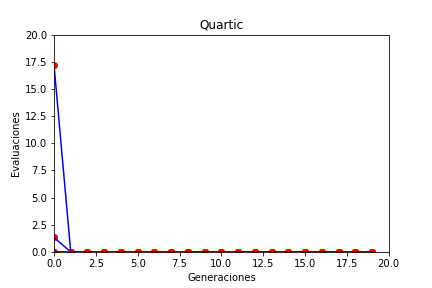
**Resultados obtenidos**

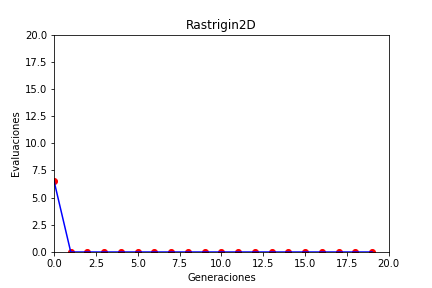


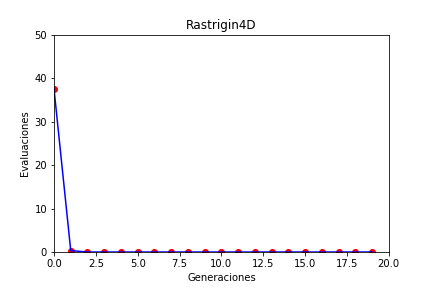
****

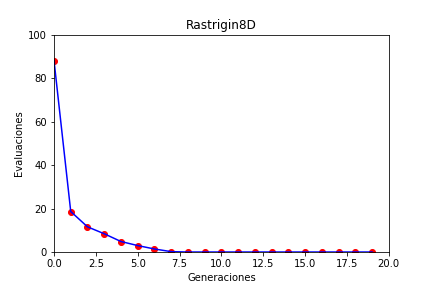
****

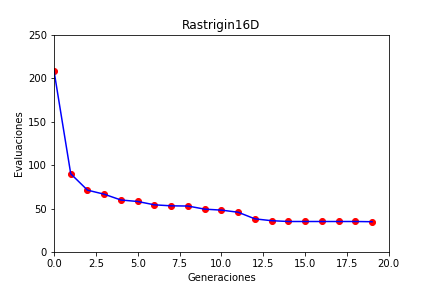
****

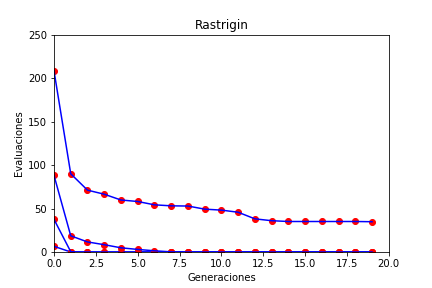
****

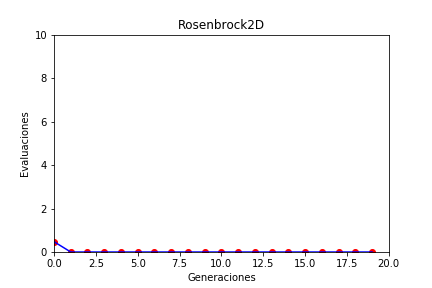
****

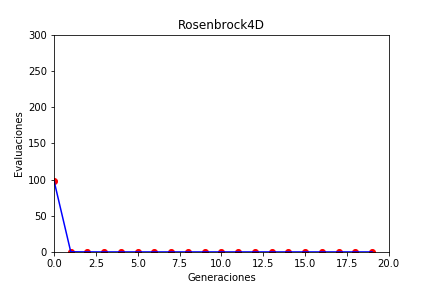
****

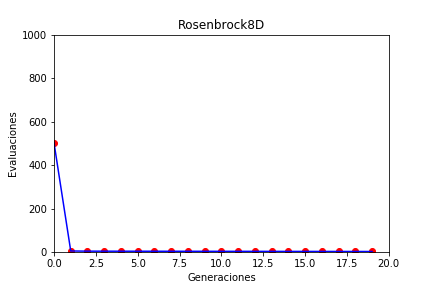
****

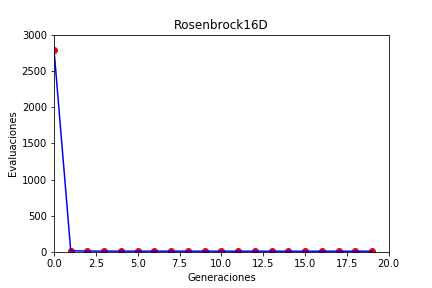
****

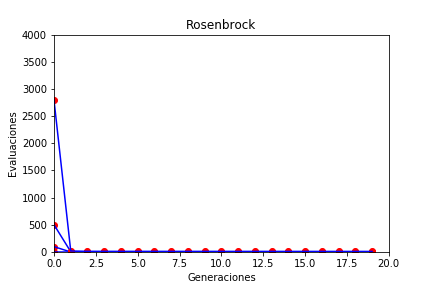
****

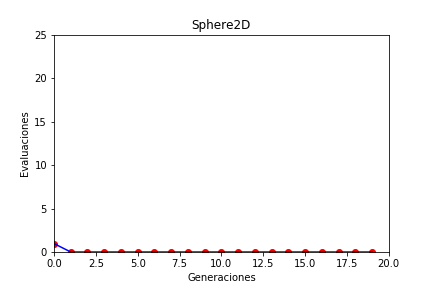
****

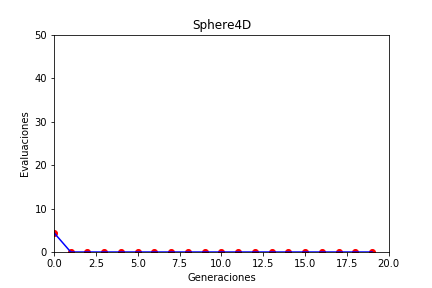
****

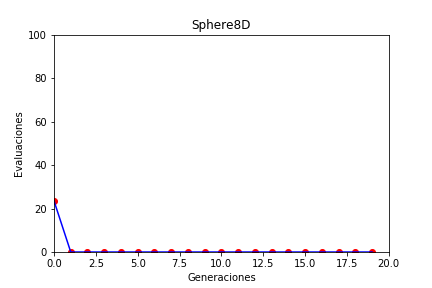
****

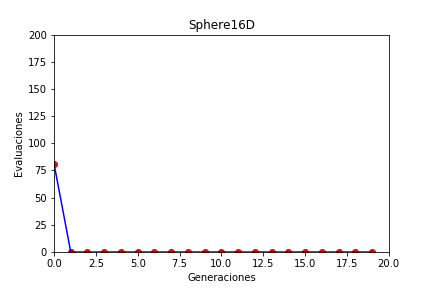
****

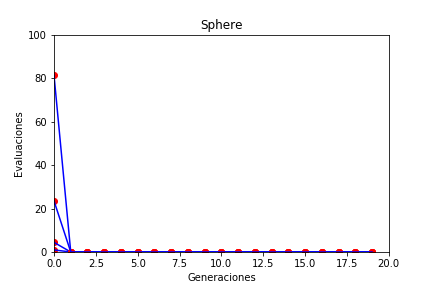
****

****

****

****

****

****

**Conclusiones**

Después de varios intentos y pruebas fallidas con errores demasiado extraños que no podía resolver, me di a la tarea de hacer el código desde cero una vez más (debido a que era bastante corto, así que no representaba una gran desventaja). Al final no sé realmente qué fue lo que cambié o arreglé, pero logré corregir los problemas, que me causaban que los fitness hicieran cosas raras y que las gráficas en lugar de solamente disminuir, de repente hicieran picos hacia arriba y luego volvieran a bajar.

Si bien esta implementación fue considerablemente sencilla, esos problemas me causaron un buen retraso, pero al final lo logré resolver.