

## PROYECTO PRÁCTICO

Se debe hacer en grupos de 3 estudiantes. Si el número total de estudiantes en la asignatura no es múltiplo de 3, hable con el profesor.

La calificación del trabajo es la misma para todo el equipo.

Si se detecta que el proyecto ha sido parcial o totalmente copiado o bajado de internet, la nota es cero.

## OBJETIVOS DE APRENDIZAJE

- Cómo se diseña e implementa un algoritmo genético (usando **Ruby**).
- Cómo se usa un algoritmo genético para resolver un problema.
- Cómo se verifica si un algoritmo genético funciona correctamente (por medio de **Cucumber**).
- Cómo se hace una pequeña investigación cambiando parámetros del problema y comparando resultados.
- Trabajo en equipo.

## PROBLEMA A RESOLVER

El trabajo consiste en escribir un algoritmo genético (GA) en paradigma OO y lenguaje Ruby, siguiendo una metodología de desarrollo ágil basada en pruebas (Cucumber), y aplicarlo para resolver un problema fácil parametrizado según el tamaño del problema N. Concretamente hay que resolver un sistema de N ecuaciones y N incógnitas:  $Ac = B$

$$\begin{pmatrix} a_{00} & a_{01} & \dots & a_{0N-1} \\ a_{10} & a_{11} & \dots & a_{1N-1} \\ \dots & \dots & \dots & \dots \\ a_{N-10} & a_{N-11} & \dots & a_{N-1N-1} \end{pmatrix} \cdot (c_0, c_1, \dots, c_{N-1}) = \begin{pmatrix} b_0 \\ b_1 \\ \dots \\ b_{N-1} \end{pmatrix}$$

A partir de unos valores concretos de A y B, el algoritmo genético debe encontrar el vector c que los satisfaga. Es decir, c es el cromosoma.

Los valores de A conviene generarlos al azar. Y luego crear también al azar un vector solución s. Por último, calcular el vector  $B = As$ . De este modo nosotros conocemos la solución s, pero el algoritmo genético tiene que descubrirla.

## METODOLOGÍA

Cada estudiante en el grupo debe asumir un rol (decídanlo entre ustedes):

- El ingeniero de pruebas debe de codificar las pruebas de aceptación en un lenguaje llamado “Gherkin” que es prácticamente igual que el español y que se usa con la herramienta *cucumber*. Debe crear al menos **4 escenarios** donde se verifique la **mutación**, el **cruce uniforme**, la **selección por torneo** y un **sistema de dos ecuaciones y dos incógnitas**.
- El ingeniero desarrollador debe de escribir el código en Ruby correspondiente, y depurarlo hasta que pase las pruebas de aceptación. Debe escribir al menos **4 clases: Gen, Cromosoma, AlgoritmoGenetico y Fenotipo**. En Fenotipo se debe implementar el problema, es decir, el sistema de ecuaciones.

- El usuario investigador debe coordinar el trabajo del ingeniero de pruebas y el ingeniero desarrollador. Debe de comunicar a ambos lo que se requiere hacer, en lenguaje español (las historias de usuario, que son los escenarios en *cucumber*) y negociar con ambos si toca hacer cambios. Cuando todo esté funcionando, el usuario investigador ejecutará el programa con diversos juegos de parámetros y hará un informe sobre lo que ve. Concretamente, debe haber dos funciones objetivo:
  - MINIMIZAR EL ERROR CUADRÁTICO. Al aplicar cada cromosoma  $c$  a las ecuaciones  $A$ , sale un resultado  $R=Ac$ , que no va a coincidir con el deseado  $B$ . La aptitud será el error cuadrático cambiado de signo: 
$$aptitud = - \sum_{i=0}^{N-1} (r_i - b_i)^2$$
. El objetivo es maximizar la aptitud, es decir, minimizar el error cuadrático. Lo ideal es que llegue a 0.
  - DIVERSIDAD: esta es nueva (no vista en clase) y no usa la fórmula anterior. Lo único que busca es maximizar la diversidad de los cromosomas. Hay muchas formas de hacerlo y les recomiendo que no se compliquen mucho la vida en ello (consulte al profesor). Puede ver algunas ideas en: <http://eplex.cs.ucf.edu/noveltysearch/userspage/#intro>. La propuesta que aparece aquí en  $O(N^2)$ , que es muy costosa. Le recomiendo alguna aproximación que sea  $O(N)$ .

## PARÁMETROS

- $N$ : el número de ecuaciones e incógnitas.
- Función objetivo empleada:
  - Sólo MINIMIZAR EL ERROR CUADRÁTICO.
  - Sólo DIVERSIDAD.
  - Una combinación entre ellas (como usted la quiera hacer, pero tiene que explicarla).

## RESULTADOS A MEDIR PARA CADA COMBINACIÓN DE PARÁMETROS

- Tiempo de ejecución hasta encontrar la solución.
- Calidad de la solución, esto es, cuanto vale el error cuadrático en el mejor cromosoma, para un tiempo fijo de ejecución (por ejemplo, 30 segundos).

## ENTREGABLES

Hay que hacer un corto documento.pdf con los resultados obtenidos en gráficas, y donde se explique el tipo de combinación entre MINIMIZAR EL ERROR CUADRÁTICO / DIVERSIDAD que se implementó. Y subirlo al campus junto con todos los archivos del programa (\*.rb, \*.features \*.pdf) en único archivo comprimido (.tgz).

Y hay que hacer una muy corta presentación en clase (5 minutos).