

Asignatura: COMPUTACIÓN EVOLUTIVA
Profesor: Ángel García Baños

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.

TRABAJO A REALIZAR

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 en el tamaño del problema N , concretamente las **N-damas**: en un tablero de ajedrez de $N \times N$ casillas, colocar N damas de modo que no se ataquen mutuamente.

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

- Usuario investigador
- Ingeniero de pruebas
- Ingeniero desarrollador

El usuario 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) y negociar con ambos si toca hacer cambios. El ingeniero de pruebas debe de codificar esa información en un lenguaje, llamado “Gherkin” que es prácticamente igual que el español, pero donde especifica las pruebas de aceptación del usuario. El ingeniero desarrollador debe de escribir el código en Ruby correspondiente, y depurarlo hasta que pase las pruebas de aceptación.

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:

- **ATAQUES**: esta es la normal de cualquier algoritmo genético, concretamente en este caso, calcula cuantas *damas* se atacan mutuamente. El objetivo es disminuir este número. Si llega a cero es que hemos encontrado la solución.
- **DIVERSIDAD**: esta es nueva y no tiene en cuenta el ataque entre *damas*. Lo único que busca es maximizar la diversidad de los cromosomas. Hay muchas formas de hacerlo y le recomiendo que no se complique 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 damas y lado del tablero.
- Función objetivo empleada:
 - Sólo ATAQUES.
 - 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.
- Calidad de la solución, esto es, cuantos ATAQUES quedan sin resolver en el mejor cromosoma.

ENTREGABLES

Hay que hacer un corto documento.pdf con los resultados medidos, que incluyan el tipo de combinación entre ATAQUES/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).