

SISTEMAS INTELIGENTES

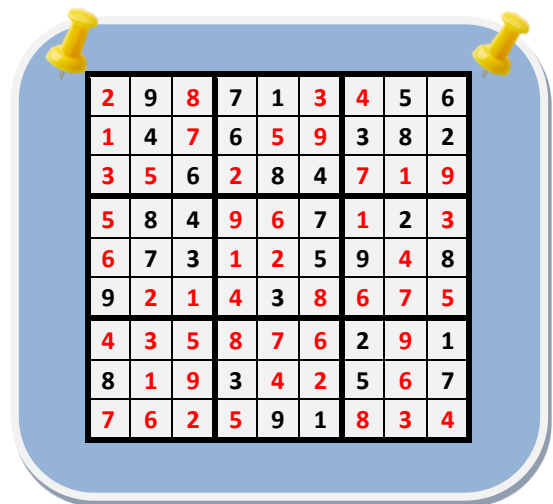
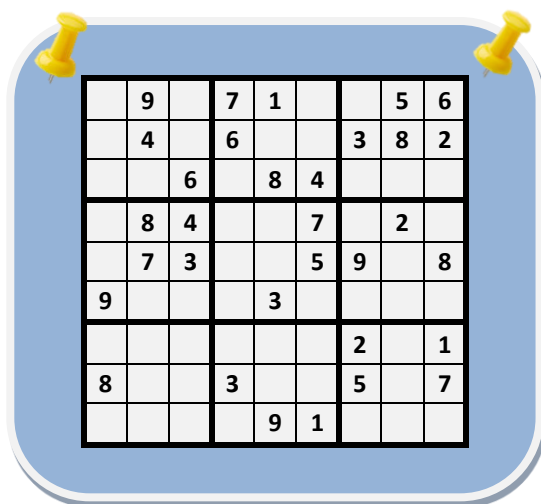
Caso de estudio

Sudoku es un juego matemático que se publicó por primera vez a finales de la década de 1970 y se popularizó en Japón en 1986, dándose a conocer en el ámbito internacional en 2005 cuando numerosos periódicos empezaron a publicarlo en su sección de pasatiempos. El objetivo del sudoku es rellenar una cuadrícula de 9×9 celdas (81 casillas) dividida en subcuadrículas de 3×3 (también llamadas “cajas” o “regiones”) con los números del 1 al 9 partiendo de algunos números (pistas) ya dispuestos en algunas de las celdas. Lo importante es que no se deben repetir en una misma fila, columna o subcuadrícula los distintos números.

Un sudoku está bien planteado si la solución es única, algo que el matemático Gary McGuire ha demostrado que no es posible si no hay un mínimo de 17 números de pista al principio.

La solución de un sudoku siempre es un cuadrado latino, aunque el recíproco en general no es cierto ya que el sudoku establece la restricción añadida de que no se puede repetir un mismo número en una subcuadrícula.

En la siguiente figura se muestra un ejemplo de posible configuración inicial junto con su solución.



Objetivo de la práctica

El objetivo de esta práctica es resolver el problema planteado proporcionando la solución. Para ello, debemos:

- Diseñar, explicar e implementar un algoritmo genético para resolver un sudoku en C++ usando el entorno de desarrollo Code::Blocks bajo Windows y la librería GAlib (deben usarse obligatoriamente los operadores genéticos proporcionados por los profesores y por la librería)
- Realizar las correspondientes pruebas de ajuste.
- Proporcionar las soluciones de distintos casos.
- Desarrollar los correspondientes documentos de explicación y necesarios para la utilización de la herramienta construida.

Elementos para diseñar, realizar y entregar

La entrega se realizará mediante una TAREA que se activará en su momento en el aula virtual SAKAI.

NOTA: Cualquier información o material que se utilice debe indicarse explícitamente su procedencia mediante las referencias bibliográficas. Si se detecta alguna información o material de alguna fuente que no se explicita, la práctica se dará directamente por “no superada”.

ELEMENTOS A ENTREGAR:

- **Documento:** Un documento pdf ([Practica1.pdf](#)) que contenga:
 - a) Explicación breve y completa de la técnica Algoritmo Genético (AG). Debe quedarse muy claro cuáles son los elementos y el proceso que sigue dicha técnica.
 - b) Explicación detallada de todas las preguntas realizadas en la sección “Cuestiones para el diseño e implementación” de este guión (en concreto, las preguntas 3), 4), 7), 9), 10) y 12)).
 - c) La tabla completa que muestre los valores de la función fitness para todas las pruebas realizadas para el ajuste del software sobre los distintos valores para los parámetros ajustables y para los distintos “Casos para el Ajuste” propuestos. La tabla debe mostrarse de forma clara y que facilite su análisis. Ver sección “Ajuste del Algoritmo Genético” en este mismo documento.
 - d) Análisis de las pruebas de ajuste (esto es, el análisis de la tabla de resultados del apartado c). El objetivo de este análisis es obtener el comportamiento del software diseñado.
 - e) A partir de este análisis, indicar qué valores de los parámetros debemos utilizar, y el protocolo para asignarlos, para que el software tenga un comportamiento aceptable. Denominar a esta sección “manual-Asignación”.
 - f) Para cada uno de los “Casos del Usuario” (ver sección correspondiente), indicar el protocolo seguido para resolverlo (siguiendo el “manual-Asignación”), el valor para cada parámetro, la solución obtenida y su fitness.
 - g) Referencias bibliográficas. Éstas deben de ser útiles para el desarrollo de la práctica y deben seguir el formato indicado. No se debe poner el material proporcionado en clase para el desarrollo de esta práctica (documentos pdf, manual GALib, etc).

Para feedback

El documento debe tener la siguiente estructura:

- i. Página 1: nombre-asignatura, curso, subgrupo, nombre-apellidos, nº-práctica/fecha-entrega
- ii. Índice del documento.
- iii. CONTENIDO
- iv. Bibliografía comentada en el formato indicado en el ANEXO I. Debajo de cada referencia debe añadirse una breve explicación, con una extensión de 3-4 líneas máximo, que informe de qué trata la referencia y para qué se utiliza en la práctica. Además, cada referencia debe estar mencionada en su correspondiente/s parte/s del documento que se haya/n usado.

NOTA: la información y explicación no contenida en este documento se entenderá como no realizada.

- **Software:**
 - h) Código fuente documentado.
 - i) Ejecutable en Windows.
El software debe ejecutarse en línea de comando teniendo como parámetros “Fichero con el caso”, “Tamaño de la población”, “Operador de Selección”, “Probabilidad de cruce” y “Probabilidad de mutación”.
 - j) Manual de uso. Debe incluir como mínimo la forma de ejecución, los parámetros que se deben proporcionar junto a una explicación de ellos y una explicación de la salida proporcionada.

- Ficheros que contengan todos los casos resueltos:
 - k) Fichero (o ficheros) en texto plano con todas las salidas obtenidas para los distintos “Casos para el Ajuste” indicando qué caso se está resolviendo, los valores de los parámetros utilizados, la salida obtenida y su valor fitness.
 - l) Fichero (o ficheros) en texto plano con todas las salidas obtenidas para los distintos “Casos del usuario” indicando qué caso se está resolviendo, los valores de los parámetros utilizados, la salida obtenida y su valor fitness.

Cuestiones para el diseño e implementación

En el fichero que denominaremos “Sudoku.cpp” debemos implementar un AG (utilizando la librería GALib) para resolver un Sudoku 9x9. En él deberás incorporar los elementos proporcionados por los profesores y otros que deberás diseñar e implementar como resultado de este trabajo.

A continuación se describen los elementos diseñados e implementados que habrá que incorporar en dicho fichero, y se pregunta sobre lo que hay que diseñar e implementar.

- Representación de un individuo

La representación que utilizaremos será un vector de enteros.

Un individuo estará representado por un vector de valores enteros de tamaño 81 que contiene de forma consecutiva las distintas filas del Sudoku. Adicionalmente, tenemos que adjuntarle la configuración de números que vienen ya dados y colocados en la configuración inicial del Sudoku. Este es el objetivo de la estructura de datos “**plantilla**” definida en el fichero “PlantillaSudoku.txt”. La lectura de los números dados del sudoku se realiza desde un fichero utilizado tanto para los “Casos para el ajuste” como para los “Casos del usuario”. En este fichero, un 0 indica que el número correspondiente a esa posición no viene dado en la configuración inicial. Incorporar el código de “PlantillaSudoku.txt” a “Sudoku.cpp”.

- 1) Añadir la definición del genoma en “Sudoku.cpp”.

- Inicialización de un individuo

El operador de inicio está definido en la función **InicioSudoku()**. Se proporciona de forma completa en el fichero “InicioSudoku.txt” salvo el casting necesario para recuperar en la variable **genome** la variable genérica **g**. Incorporar el código de “InicioSudoku.txt” a “Sudoku.cpp”.

- 2) Añadir el casting en el sitio indicado en la función **InicioSudoku()**.

- 3) Dada esa definición, explica de qué manera se están inicializando los individuos en el AG propuesto.

- Operadores genéticos

- Selección

Dada la representación de los individuos para resolver este problema.

- 4) Explica el funcionamiento de los operadores de selección indicados en la sección “Ajuste del Algoritmo Genético”.

- 5) Insértalos de la manera adecuada en “Sudoku.cpp”.

- Cruce

El operador de cruce está definido en la función **CruceSudoku()** proporcionado en el fichero "CruceSudoku.txt". La implementación se proporciona de forma completa salvo el casting necesario para recuperar en las variables m, p, h1 y h2 las variables p1, p2, c1 y c2 respectivamente. Incorporar el código de "CruceSudoku.txt" a "Sudoku.cpp".

6) Escribir el casting en el sitio indicado en la función **CruceSudoku()**.

7) Dada esa definición, explica de qué manera se están cruzando los individuos.

- Mutación

El operador de mutación está definido en la función **MutacionSudoku()** proporcionado en el fichero "MutacionSudoku.txt". La implementación se proporciona de forma completa salvo el casting necesario para recuperar en la variable genome la variable genérica g. Incorporar el código de "MutacionSudoku.txt" a "Sudoku.cpp".

8) Escribir el casting en el sitio indicado en la función **MutacionSudoku()**.

9) Dada esa definición, explica de qué manera se están mutando los individuos.

- Condición de parada

10) Define y explica la condición de parada que utilizarás.

11) Implementar la condición de parada en "Sudoku.cpp".

- Función fitness

12) Diseña y explica la función fitness que utilizarás. Recuerda, como se indica al comienzo de este guión, que una solución del sudoku no puede repetir en una misma fila, columna o subcuadrícula ninguno de los números

13) Implementar la función fitness en "Sudoku.cpp".

- Completar con la implementación de los elementos necesarios para cubrir los objetivos del AG propuesto. NOTA: obligatoriamente utilizaremos en valor 1 en el método "evolve" (**evolve(1)**).

Ajuste del Algoritmo Genético

Para el ajuste de parámetros se utilizará el software desarrollado con ejecución en línea de comando. Cuando lo ejecutemos con distintos valores de parámetros se creará un "script" para tal efecto.

Para el ajuste utilizaremos los siguientes valores para los distintos parámetros.

Parámetros fijos:

Número de Generaciones = 12.000

Operador de cruce = CruceSudoku

Operador de mutación = MutacionSudoku

Parámetros con valores:

Tamaño de la población – {100,150}

Operador de Selección – {GARouletteWheelSelector, GATournamentSelector}

Probabilidad de cruce – {0.8, 0.85, 0.9, 0.95}

Probabilidad de mutación – {0.01,0.05, 0.1, 0.125, 0.15}

"Casos para el Ajuste": Caso-A1, Caso-A2, Caso-A3, Caso-A4, Caso-A5 (disponibles en recursos del aula virtual).

Casos del Usuario

Para la prueba final del sistema construido junto al ajuste, debemos resolver distintos casos desde el punto de vista del usuario final. Para ello, utilizaremos los casos Sudoku-1, Sudoku-2, Sudoku-3 disponibles en recursos del aula virtual.

Evaluación

En la evaluación de la práctica se tendrán en cuenta los siguientes criterios:

Documento Practica1.pdf

Puntos: 6

La no entrega de este documento supondrá la NO superación de la práctica.

Se valora que sea entendible, que esté bien estructurado, que sea completo (que contenga la información indicada anteriormente y que el alumno considere relevante). Y cualquier información que refleje el trabajo realizado y que sirva para una mejor comprensión del mismo.

Apartado a)	0.5 puntos
Apartado b)	2.75 puntos
Apartado c) y d)	1.75 puntos
Apartado e) y f)	1.0 puntos
Bibliografía	0.25 puntos (extra)

Manual de Uso

La no entrega supondrá la **reducción de 1 punto** en la calificación final de la práctica. La corrección y utilidad será evaluado en el apartado de software.

Ficheros de las pruebas

La **no entrega de “todos los ficheros”** sobre las distintas pruebas indicadas tanto en el apartado j) como en k) **supondrá una reducción de 1.5 puntos** en la calificación final de la práctica.

Software

Puntos: 4

La no entrega del código fuente o el no funcionamiento de la herramienta software (se queda bloqueado, no lee los datos,...) **supondrá la NO superación** de la práctica.

Si no se entrega el ejecutable pero este puede generarse a partir del código fuente entregado sin ningún problema supondrá una **reducción de 1 punto** en la calificación final.

Partiendo de que el software funciona correctamente¹, se valora la interpretabilidad de la salida proporcionada (1 punto), la calidad de las soluciones proporcionadas (1.5 puntos), que el código esté correctamente documentado y que el manual de uso sea correcto y de utilidad para el usuario (1 punto), y que sigue las especificaciones indicadas para la ejecución (0.5 puntos).

¹ el software cumple con las especificaciones para las que ha sido construido y logra su cometido.

ANEXO**FORMATO DE LA BIBLIOGRAFÍA PARA LA PRÁCTICA:**

- Se usará el formato que se indica a continuación.
- Si tuviera algún otro documento no especificado en los ítems mostrados, por favor, adapte algunos de ellos a ese nuevo documento.
- Bibliografía ordenada por orden alfabético del apellido del primer autor.
- Citarlas en el texto con el número correspondiente entre corchetes.
- Debajo de cada referencia debe añadirse una breve explicación, con una extensión de 3-4 líneas máximo, que informe de qué trata la referencia y para qué se utiliza en la práctica.

Las referencia [1] se refiere a un manual de alguna herramienta o sistema (en texto o en documento electrónico). La referencia [2] se refiere a un artículo de revista. Las referencias [3,9] se refieren a artículos de congresos. Las referencias [4,6] se refieren a capítulos o partes de un libro. La referencia [7] se refiere a un libro. Las referencias [5,8] se refieren a documentos electrónicos y páginas web.

REFERENCIAS

- [1] M. Achour, F. Betz and, A. Dovgal (2013). "Manual de PHP" [Online]. Available: <http://www.php.net/manual/es/index.php>
- [2] A. Castaldini, A. Cavallini, B. Fraboni, P. Fernandez, and J. Piqueras (1997). "Midgap traps related to compensation processes in CdTe alloys", *Physical Review B*, vol. 56, no. 23, pp. 14897--14900.
- [3] R. K. Gupta and S. D. Senturia (1997). "Pull-in time dynamics as a measure of absolute pressure". In *Proceedings IEEE International Workshop on Microelectromechanical Systems (MEMS'97)*, Nagoya, Japan, pp. 290--294.
- [4] D. E. Knuth (1973). "Fundamental Algorithms (2nd ed.), The Art of Computer Programming". Reading, Massachusetts: Addison-Wesley, vol. 1, section 1.2, pp. 10--119.
- [5] P. Lehman (2007). The biblatex package [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/exptl/biblatex/>
- [6] D. D. Lincolll (1977). "Semigroups of recurrences," in *High Speed Computer and Algorithm Organization*, 3rd ed., Fast Computers, D. J. Lipcoll, D. H. Lawrie, and A. H. Sameh, Eds. New York: Academic Press, no. 23, part 3, pp. 179--183.
- [7] S. M. Metev and V. P. Veiko (1998). "Laser Assisted Microtechnology" 2nd ed., Ed. Berlin, Germany: Springer-Verlag.
- [8] M. Shell (2007). IEEEtran webpage on {CTAN} [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran/>
- [9] M. Yajnik, S. B. Moon, J. Kurose, and D. Towsley (1999). "Measurement and modeling of the temporal dependence in packet loss", in *Proceedings IEEE INFOCOM'99*, vol. 1, New York, NY, pp. 345--352.