

ACTIVIDAD 1

(NOTA IMPORTANTE: Queda prohibido usar contenido generado por herramientas de Inteligencia Artificial Generativa para la realización de cualquier parte de esta actividad.)

1. PRERREQUISITOS

La presente actividad deberá ser realizada tras el estudio de los temas 1, 2, 4, 9 y 14 del libro de texto base de la asignatura: “Fundamentos de la Computación Evolutiva” de Carmona y Galán, editorial Marcombo, año 2020. Se debe estudiar el tema 1, ya que proporciona al alumno una visión general e introductoria del funcionamiento de un algoritmo evolutivo. Se debe estudiar a su vez el tema 2, porque describe las características fundamentales de un algoritmo genético, que es la variante de algoritmo evolutivo en la que se centra esta actividad. En este sentido, recomendamos leer también el tema 4, ya que la programación evolutiva clásica ofrece una perspectiva complementaria a la representación de individuos (mediante máquinas de estados finitos) y, además, la programación evolutiva moderna se acerca conceptualmente a las estrategias evolutivas, de las que se ocupará la Actividad 2. Adicionalmente, es necesario estudiar el tema 14, puesto que entra en detalle sobre el funcionamiento interno de un algoritmo genético: por qué consigue optimizar y cómo se comporta ante problemas de diferente complejidad. Por último, se debe estudiar el tema 9, ya que ofrece herramientas y da pautas para la correcta evaluación de un algoritmo evolutivo, en general, y de un algoritmo genético, en particular.

2. DEFINICIÓN DEL PROBLEMA

En esta actividad se aborda el *Problema del Sudoku*. Un sudoku es un juego matemático en el que hay que rellenar una cuadrícula de 9x9 celdas, dividida en subcuadrículas de 3x3, con las cifras del 1 al 9. El juego está sujeto a la restricción de que no se puede repetir ningún número en cada fila de la cuadrícula, en cada columna de la cuadrícula o en cada subcuadrícula. Inicialmente, algunas de las celdas están ya rellenas, de manera que la solución al sudoku sea siempre única. La figura 1 muestra un ejemplo de sudoku y su solución.

	6		1		4		5	
		8	3		5	6		
2								1
8			4		7			6
		6				3		
7			9	1				4
5								2
		7	2		6	9		
	4		5		8		7	

9	6	3	1	7	4	2	5	8
1	7	8	3	2	5	6	4	9
2	5	4	6	8	9	7	3	1
8	2	1	4	3	7	5	9	6
4	9	6	8	5	2	3	1	7
7	3	5	9	6	1	8	2	4
5	8	9	7	1	3	4	6	2
3	1	7	2	4	6	9	8	5
6	4	2	5	9	8	1	7	3

Figura 1: Sudoku (izquierda) y su solución (derecha), tomados de “Algoritmos Evolutivos: Un Enfoque Práctico” de L. Araujo y C. Cervigón, Editorial Ra-Ma, Año 2009.

3. IMPLEMENTACIÓN

De cara a poner en práctica los conocimientos adquiridos en los temas mencionados en la sección 1 de “Prerrequisitos”, se pide programar **desde cero** un algoritmo genético para la resolución del problema del sudoku. Se puede definir un problema del sudoku particular a partir de un fichero de texto donde figure el contenido de las 81 cuadrículas, incluidas las vacías. Por otra parte, la especificación de los parámetros que necesita el algoritmo genético para su funcionamiento es recomendable realizarla a través de otro fichero de texto adicional.

El algoritmo genético pedido deberá tener las siguientes características específicas:

(1) Representación de cada individuo

Se utilizará una representación que asocie un gen a cada celda. Mediante un valor *entero* en cada gen (comprendido entre 1 y 9) se establecerá el contenido de la celda correspondiente. Por tanto, el genotipo puede ser considerado como una cadena de enteros, aunque nosotros dejamos libertad al alumno para utilizar aquella estructura de datos que considere más conveniente para implementar el genotipo recién descrito.

(2) Función de adaptación (o función fitness)

La función fitness asociará a cada individuo un valor numérico calculado del siguiente modo:

$$\frac{\sum_{i=1}^{81} \{f(i) + c(i) + s(i)\}}{2},$$

donde $f(i)$ es el número de celdas en la misma **fila** que la celda i cuyo valor coincide con el de esta última celda, $c(i)$ es el número de celdas en la misma **columna** que la celda i cuyo valor coincide con el de esta última celda y $s(i)$ es el número de celdas en la misma **subcuadrícula** que la celda i cuyo valor coincide con el de esta última celda. Nótese que la solución al sudoku (mejor caso) tendrá un valor fitness igual a cero, mientras que una hipotética cuadrícula donde las 81 celdas contuvieran la misma cifra (peor caso) tendrá un valor fitness de 972.

(3) Inicialización de la población

Se realizará de forma *aleatoria*, asignando a cada celda vacía de un individuo un número de entre {1, ..., 9} con igual probabilidad. No obstante, de cara a reducir significativamente el tamaño del espacio de búsqueda, se recomienda tener en cuenta las restricciones iniciales del sudoku determinadas por sus celdas rellenas. Es decir, una vez que se define un problema del sudoku particular, cada celda vacía sólo tiene un subconjunto de valores dentro de {1, ..., 9} que podrá tomar para cumplir con todas las restricciones del sudoku. Para una celda vacía particular, ese subconjunto podría ser, por ejemplo, {3, 6} y entonces es recomendable que la inicialización asigne valor a dicha celda eligiendo un valor aleatorio únicamente de entre {3, 6}.

(4) Selección de padres

Se realizará mediante *selección por torneo* (véase [Carmona y Galán 2020, sección 2.4.3]).

(5) Cruce

Se realizará mediante *cruce por un punto* (véase [Carmona y Galán 2020, sección 2.5.1.1]) sobre las cadenas de enteros asociadas a los genotipos de los dos padres, con una *probabilidad de cruce* dada (véase [Carmona y Galán 2020, segundo párrafo de la sección 2.5]).

(6) Mutación

Se realizará sobre cada gen de cada individuo asignando al gen correspondiente, con una *probabilidad de mutación* dada (véase [Carmona y Galán 2020, segundo párrafo de la sección 2.6]), un valor diferente al actual que se calcula aleatoriamente con distribución uniforme sobre el rango de valores del gen (exceptuando el valor actual). No obstante, la misma recomendación

descrita para el caso de la inicialización de la población es aplicable también para la mutación, de cara a reducir significativamente el tamaño del espacio de búsqueda. Es decir, es recomendable que todo nuevo valor de celda generado tras la mutación cumpla todas las restricciones del sudoku inicial.

(7) Selección de supervivientes

Se utilizará un *modelo generacional*, que reemplaza toda la población actual por otra nueva (véase [Carmona y Galán 2020, sección 2.7]). Además, se aplicará *elitismo* (véase [Carmona y Galán 2020, segundo párrafo de la sección 2.7]), de modo que el mejor individuo en la población actual es preservado en la siguiente generación, siempre que no haya un individuo en la nueva población con igual o mejor valor de adaptación. En este caso, el individuo a reemplazar en la nueva población puede ser el de menor valor de adaptación o puede ser elegido aleatoriamente.

4. EVALUACIÓN EXPERIMENTAL

Una vez programado el algoritmo genético, se pide evaluarlo partiendo de las siguientes premisas:

(a) Problemas de prueba

La idea es emplear problemas del sudoku de diferente complejidad; por ejemplo, bastaría con un problema sencillo y otro complejo. Por regla general, en problemas sencillos el algoritmo genético será más eficiente (en cuanto a tiempo de ejecución) si aplica más explotación, mientras que en problemas complejos será más eficaz (en cuanto a la calidad de la mejor solución encontrada) si aplica más exploración.

El alumno puede utilizar instancias del problema del sudoku que encuentre en internet o, de forma alternativa, utilizar las dos instancias mostradas en la figura 2, **sencilla** y **compleja** respectivamente:

				3				4
	9		4		6		7	
	5					3	8	
				7	8			3
3						6	9	
5	4		6				2	
7		5		2	4			
9	8	4		6	5	2		
	2	6		8				9

	3	8					6	
5	4					2		
		1		9		3		
8	6			4			7	
4				8	3	5		
		7					4	
3	8	6			9			4
	2		4	6			3	
					7		8	

Figura 2: Instancias sencilla (izquierda) y compleja (derecha) del problema del sudoku, tomadas de <http://www.sudoku-online.org/> dentro de las categorías “fácil 10 (#1217)” y “difícil 10 (#3217)” respectivamente.

En último término se trata de emplear dos tipos de instancia del problema del sudoku: una sencilla de resolver (que produzca mejores resultados al aplicar mayor explotación) y otra difícil de resolver (que produzca mejores resultados al aplicar mayor exploración).

(b) Elección de parámetros del algoritmo genético

Cuando sea posible, el alumno utilizará valores para los parámetros del algoritmo genético recomendados en el texto base de la asignatura o en la literatura. En cualquier caso, de cara a

lograr cierta homogeneidad en la implementación de los alumnos, se recomienda fijar el tamaño de población en 10 individuos para la instancia sencilla y en 100 individuos para la instancia compleja del problema del sudoku. (Estos valores pueden ser variados por los alumnos si así lo creen conveniente para su implementación, de cara a obtener resultados lo más ilustrativos posible: que no siempre lleguen al óptimo global, en un extremo, ni que se comporten de forma parecida a una búsqueda aleatoria, en otro extremo.)

(c) Metodología de evaluación

Esta actividad requiere evaluar el algoritmo genético utilizando las medidas y tipos de gráficas que aparecen en el capítulo 9 del texto [Carmona y Galán, 2020]. Por ejemplo, la tasa de éxito o las curvas de progreso son dos herramientas útiles para ilustrar el funcionamiento del algoritmo genético, junto con otras que aparecen en el capítulo 9 y que el alumno pueda considerar necesarias. También puede resultar de interés el medir el tiempo de ejecución empleado. Para que todas estas medidas sean significativas estadísticamente, se recomienda tomar medias para un número suficientemente amplio de ejecuciones (repetidas para una misma instancia del problema de optimización y para un mismo conjunto de parámetros del algoritmo).

Se pide, tanto para la instancia sencilla como para la instancia compleja del problema del sudoku, **evaluar el algoritmo genético en función de la probabilidad de mutación** empleada en la fase de mutación.

5. CRITERIOS DE CALIFICACIÓN

Por último, los criterios de calificación de esta actividad son los siguientes:

- **Programación del algoritmo genético** (4 puntos sobre 10)

El alumno deberá entregar tanto el código fuente como el ejecutable. Se valorará la correcta estructuración del código, la programación de forma modular y la inclusión de comentarios adecuados. Por otra parte, de cara a que el código desarrollado en esta actividad puede ser reutilizado y ampliado en futuras actividades o trabajos, se valorará también que se separen claramente aquellas partes que pueden ser comunes a cualquier algoritmo evolutivo (por ejemplo, la selección de padres y la selección de supervivientes) de aquellas que son propias del problema o representación actuales (por ejemplo, la mutación).

- **Evaluación del algoritmo genético** (4 puntos sobre 10)

La principal pregunta que hay que responder en este apartado es qué relación existe entre la probabilidad de mutación y la eficacia (y eficiencia) del algoritmo genético, es decir, cómo influye la probabilidad de mutación en la calidad (y en la velocidad de convergencia) de los resultados obtenidos tanto para una instancia sencilla del problema del sudoku como para otra compleja. Dicho de otro modo, ¿es conveniente variar el grado de exploración para una instancia compleja respecto a otra sencilla? Para dar respuesta a esta pregunta, el alumno deberá apoyarse en los experimentos que realice y en los resultados que obtenga de los mismos.

- **Redacción de la memoria de la actividad (o informe técnico)** (2 puntos sobre 10)

El alumno deberá entregar la memoria de la actividad en formato PDF a través del curso virtual de la asignatura. La memoria debería contener al menos las siguientes secciones: “Introducción”, “Descripción del Problema del Sudoku”, “Implementación”, “Evaluación Experimental” y “Conclusiones”. Se valorará la redacción clara, la presentación de los resultados experimentales de una forma ilustrativa y la discusión de los mismos de una manera lógica.