LEVANTAMENTO BIBLIOGRÁFICO DAS TÉCNICAS DE PROCESSAMENTO DE IMAGEM PARA RESOLUÇÃO DO SUDOKU

ARTÍCULO CIENTÍFICO

RECIBIDO: 15/07/2019 ACEPTADO: 15/09/2019 ÚLTIMA VERSIÓN: 15/02/2020

BIBLIOGRAPHIC SURVEY OF IMAGE PROCESSING TECHNIQUES FOR SUDOKU RESOLUTION

[1] João Marcos de Andrade

[2] Me. Wesley Pereira Marcos

Como citar este artículo: Andrade, J; Pereira, W, (2020) Levantamiento bibliografico das técnicas de processamento de imgem para resolicao do sudoku, revista científica Dynamikos 1 (1). Recuperado de: https://teinco.edu.co/revistas-2-2/

RESUMEN

Sudoku es un juego lógico y un problema matemático muy famoso que requiere ciertas rutinas de resolución. Cuando se utiliza un ordenador para resolverlo es necesario transponer el juego del papel al ordenador, donde el procesamiento de imágenes entra para realizar una tarea de este tipo. Así, el presente estudio tiene como objetivo estudiar la bibliografía sobre algoritmos para la resolución del juego con la ayuda del procesamiento de imágenes, además de realizar una ligera comparación entre los algoritmos más utilizados.

PALABRAS CLAVE: Sudoku, Procesamiento de imágenes, Sudoku Solver, Backtracking, Algoritmos

ABSTRACT

Sudoku is a logical game and a very famous mathematical problem which demands certain solving routines. When using a computer to solve it, it is necessary to transpose the paper game to the computer, where image processing enters to perform this task. Thus, the present study aims at a bibliographic survey on algorithms for solving the game with the aid of image processing, in addition to performing a slight comparison between the most used algorithms.

KEYWORDS: Sudoku, Image processing, Sudoku solver, Backtracking, Algorithms.

Sumario: 1. Introducción. 2. Método. 2.1. Imagen digital. 2.1.2. Muestreo y cuantificación de imágenes. 2.1.3. Representación de imágenes digitales. 2.1.4. Herramientas utilizadas en el procesamiento de imágenes. 2.2. Sudoku 2.2.2. Historia 2.2.3. Reglas del juego 2.2.4. Algoritmo de solución 2.3. Trabajos relacionados 3. Resultados. 4. Discusión y conclusiones. 5. Referencia bibliográfica.
Contents: 1. Introduction. 2. Method. 2.1. Digital image. 2.1.2. Image sampling and quantization. 2.1.3. Representation of digital images. 2.1.4. Tools used in image processing. 2.2. Sudoku 2.2.2. History 2.2.3. Rules of the game 2.2.4. Solution algorithm 2.3. Related works 3. Results. 4. Discussion and conclusions. 5. Bibliographic reference.

1 Estudiante del Curso de Ingeniería Eléctrica - Centro Universitario de Patos de Minas, UNIPAM, Brasil– joaomarcos1@unipam.edu.br

2 Profesor, Departamento de Ingeniería Eléctrica - Centro Universitario de Patos de Minas, UNIPAM, Brasil– wesleypm@unipam.edu.br

1. INTRODUCCIÓN

La visión es el más avanzado de nuestros sentidos, por lo que no es de extrañar que las imágenes jueguen el papel más importante en la percepción humana. Sin embargo, a diferencia de las máquinas, que cubren casi todo el espectro electromagnético (EM), que van desde el rango hasta las ondas de radio, la visión humana se limita a la banda visual del espectro EM (GONZALEZ y WOODS, 2008).

Las máquinas pueden operar en imágenes generadas por fuentes que los humanos no están acostumbrados a asociar imágenes. Esto incluye ultrasonido, microscopía electrónica e imágenes generadas por computadora. Así, el procesamiento de imágenes digitales abarca un amplio y variado campo de aplicaciones (GONZALEZ y WOODS, 2008).

El área de procesamiento de imágenes se ha desarrollado significativamente en las últimas décadas. Dado que el crecimiento se observa en el área académica, donde el tema es objeto de investigación, tesis y tesis en las universidades más importantes de Brasil y el mundo. Y con el tiempo aumentó el número de empresas que utilizan soluciones de procesamiento de imágenes en sus segmentos (NETO y FILHO, 1999). Una posible aplicación es para la resolución del sudoku de una manera computacional.

Sudoku es un rompecabezas donde el jugador necesita llenar los espacios vacíos de una cuadrícula de 9×9 con números del 1 al 9, con la condición de que cada dígito aparezca una sola vez, en todas las filas, columnas y regiones 3 × 3, respetando un conjunto de espacios previamente llenos, llamados pistas (FELISBERTO et al, 2016).

Este artículo tiene como objetivo llevar a cabo una encuesta bibliográfica sobre el procesamiento de imágenes juntos de algoritmos de resolución de sudoku.

2. MÉTODO

Para entender el procesamiento de imágenes, primero hay que entender qué es una imagen, de qué está compuesta y similar. Como describe González y Woods (2018), una imagen se puede definir como una función bidimensional f(x,y), donde x e y son las coordenadas del plano, y la amplitud de 'f', en cualquier coordenadas que se llame el nivel gris en el punto dado. Es decir, cuando los valores de 'x', 'y' y 'f' son finitos y en cantidades discretas, la imagen se llama como imagen digital.

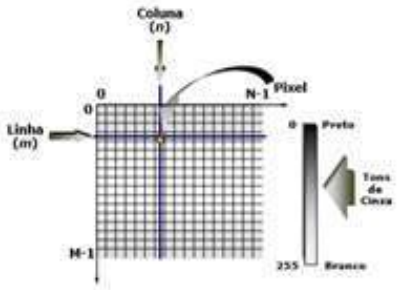
2.1 Imagen digital

Una imagen digital es una herramienta útil para los estudios, ya que puede ser la imagen de un espectro que el ser humano no es capaz de visualizar, como fuentes ultrasónicas y electrones microscópicos. Con este procesamiento de imágenes nos proporciona varias aplicaciones en diferentes campos de la ciencia (GONZALEZ E WOODS, 2008).
Como afirman Solomon y Breckon (2010), el campo del procesamiento de imágenes digitales se refiere al procesamiento de imágenes digitales por un ordenador digital. El procesamiento digital de imágenes implica procesos, cuyos inputs y salidas son imágenes y, además, implica procesos de extracción de atributos de imagen e incluso el reconocimiento de objetos individuales, como describe González y Woods (2008).

El interés en los métodos de procesamiento de imágenes proviene de dos áreas de aplicación principales. La mejora de la información visual para la interpretación humana y el procesamiento de datos de imágenes para almacenamiento, transmisión y representación, considerando la percepción automática por máquinas, como menciona Gonzalez et al (2003).

Queiroz y Gomes (2006) dicen que "debido a que las computadoras no son capaces de procesar imágenes continuas, sino sólo matrices de números digitales, es necesario representar las imágenes como arreglos bidimensionales de puntos". Por lo tanto, cada elemento que da como resultado la imagen digital se denomina píxel

(Figura.1).
Figura 1- Representación de una imagen bidimensional.



Fuente: Queiroz e Gomes, 2006.

Como se puede ver en la Figura 1, es una matriz común para localizar un píxel en una imagen. Se puede señalar "[...] el sentido de lectura (escaneo) y la convención generalmente adoptada en la representación espacial de una imagen digital." (QUEIROZ y GOMES, 2006).
González y Madera (2002) complementa al decir que la descomposición de la intensidad luminosa (Figura 1) en el punto (x,y) en: (i) componente de iluminación, i(x,y), que se entiende por la cantidad de luz incidente en el punto; y el componente de reflectancia, r(x,y), que se refiere a la luz reflejada por el punto. El Producto de i(x,y) y r(x,y) resultando en:

$$f(x,y)=i(x,y)*r(x,y) \qquad (1)$$

Dónde, y, depender de los matices de la fuente de iluminación, ya depende de las propiedades de las superficies de los objetos.$0<i(x,y)<\infty 0<r(x,y)<1 i(x,y) r(x,y)$

Un elemento importante en la creación de una imagen es el color, y los ordenadores utilizan el sistema RGB (rojo-verde-azul) en el procesamiento, donde se controla la intensidad de los colores primarios (SPECHT et al, 2013). Según González y Woods (2003) y González et al (2004), en el sistema RGB, el valor (0.0.0) corresponde al color negro, absorción total, lo que significa que es la intensidad cero en los tres componentes, y el valor (255.255.255) indica el color blanco, la reflexión total.

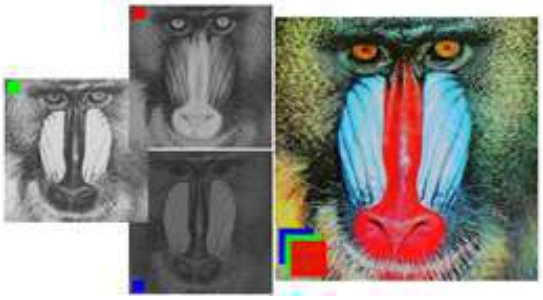
Según Queiroz y Gomes (2006), una imagen digital a color en el sistema RGB, las intensidades de los colores rojo, verde y azul son componentes vectoriales que son lo que llamamos el píxel. Así, la combinación de tres imágenes monocromáticas que forman la imagen en color que se puede ver:

$$f(x,y) = fR(x,y) + fG(x,y) + fB(x,y) \quad (2)$$

Donde, representan las intensidades de luz de los tres componentes primarios. en un momento dado.

fR(x,y),fG(x,y) e fB(x,y)

Figura 2 - Representación de una imagen digital bidimensional.



Fuente: Queiroz e Gomes, 2006.

La Figura 2 muestra los planos monocromos de una imagen y el resultado de la composición de los tres planos. Los mismos conceptos formulados para una imagen digital monocroma se aplican a cada plano de una imagen a color (GONZALEZ y WOODS, 2002).

2.1.2 Muestreo y cuantificación de imágenes

Mediante varios métodos se puede obtener una imagen, pero para realizar el procesamiento y almacenamiento de imágenes digitales en un ordenador es necesario realizar su discretización a niveles de coordenadas espaciales y brillo, donde la discretización de las coordenadas espaciales y el brillo se denominan respectivamente muestreo y cuantificación (GONZALEZ y WOODS, 2002).

Queiroz y Gomes (2006) describen que normalmente ambos procedimientos son uniformes, lo que implica el muestreo de la imagen f(x,y) en puntos equidistantes y cuya forma es una matriz M x N, donde cada elemento es una aproximación del nivel gris en el punto muestreado de la imagen a un valor en el conjunto 0, 1, ..., L-1, como se puede observar en la Figura 3.

Figura 3 - Representación de la matriz M x N.

$$F \approx \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(0,M) & f(1,M) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Fuente: Queiroz e Gomes, 2006.

Asociar el límite inferior (0) del rango de nivel de gris de un píxel, por lo tanto, el color negro y el límite superior (L-1) al blanco. Así, los píxeles que están entre 0 y L-1 presentaron diferentes tonos de gris donde más cerca de cero más oscuro estará la sombra (GONZALEZ y WOODS, 2002).

Las características de la imagen, sus dimensiones y la complejidad de los objetivos contenidos en ella, en cuanto a la aplicación a la que está destinada, pueden proporcionar el número de muestras y niveles de gris necesarios para la representación de la imagen digital de calidad adecuada. (QUEIROZ y GOMES, 2006). Figura 4 de (A) a (D) este concepto se ejemplifica.

Figura 4 – Influencia de la variación del número de muestras y niveles de cuantificación en la calidad de una imagen digital: (A) 200 x 200 píxeles / 256 niveles; (B) 100 x 100 píxeles / 256 niveles; (C) 25 x 25 píxeles / 256 niveles; y (D) 200 x 200 píxeles / 2 niveles.

Fuente: Queiroz e Gomes, 2006.

Al obtener una imagen de calidad de televisión, se necesitan 512 x 512 píxeles y 128 niveles de gris. La cantidad de niveles de ceniza considerados suficientes para la visión humana es de 64 niveles. Sin embargo, la mayoría de los sistemas de visión artificial utilizan imágenes con 256 niveles de ceniza (FILHO y NETO, 1999).

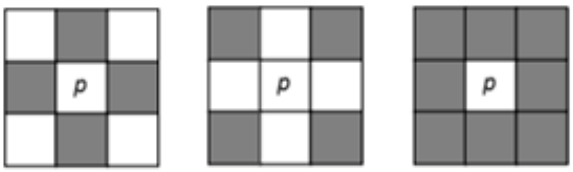
2.1.3 Representación de imágenes digitales

Como se describe en el artículo anterior, González y Woods (2018) describen una imagen como una función de f(x,y), es decir, una función bidimensional. Por lo tanto, hay a su vez una relación entre los píxeles de una imagen digital que puede influir en su procesamiento. Como factores de vecindad, adyacencia, conectividad, distancia, regiones y fronteras.

El factor de vecindad de un píxel es ejemplificado por Filho y Neto (1999) como, los píxeles que están directamente al lado del píxel p(x,y), es decir, el píxel p presenta 4 vecinos, siendo 2 horizontales y 2 verticales que las coordenadas son: (x+1,y), (x-1,y), (x,y-1) y (x,y+1).

Estos píxeles son el '4-vecindario' del píxel 'p' mencionado anteriormente, el conjunto. Sin embargo, los píxeles que se encuentran en la diagonal del píxel 'p', los que se encuentran en las coordenadas:(x+1,y+1), (x+1,y-1), (x-1,y+1) y (x-1,y-1) constituyen una vecindad junto a la '4-vecindario', que se llama '8-neighborhood', o bien el conjunto . González y Madera (2008) añade que a veces algunos puntos de este barrio no existirán porque el píxel 'p' está situado en el borde de la imagen digital. Como se puede ver en la Figura 5.

Figura 5 - Representación gráfica de 4-vecindad, vecindad diagonal y 8-vecindad, respectivamente.



Fuente: Filho e Neto, 1999.

El factor de adyacencia es considerado por González y Woods (2008) como 3 tipos, '4-adyacencia', '8-adyacencia' y 'm-adyacencia' (adyacencia mixta). Cuando hacer tal distinción es necesario considerar 'V' siendo este el valor de intensidad utilizado para definir la adyacencia entre píxeles.

Por ejemplo, puede utilizar una imagen binaria en la que el valor de V - {1} , si hace referencia a la adyacencia del valor 1 entre píxeles. En una escala de grises, se sigue el mismo razonamiento, y 'V' presentará una mayor variedad de valores (GONZALEZ y WOODS, 2008).

Teniendo en cuenta dos píxeles 'p' y 'q', que se someterán a una comprobación de adyacencia, donde si ambos con valores contenidos en 'V' serán '4-adyacentes' si 'q' pertenece a. Lo mismo ocurre con '8-adyacencia' tanto con los valores contenidos en 'V' como si 'q' pertenece a.

Sin embargo, la adyacencia mixta o 'm-adyacencia' como los otros dos tipos requieren que 'p' y 'q' tengan valores contenidos en 'V' y 'q' para pertenecer, pero también 'q' puede pertenecer a y el conjunto de ningún valor que pertenezca a 'V' (GONZALEZ y WOODS, 2008).

Analizar un subconjunto de píxeles de una imagen donde los píxeles 'p' y 'q' se dirán como conectados en la condición de una "ruta" de píxeles adyacentes que pertenecen al mismo subconjunto existen entre ellos (FIGURA 6). Cuando cualquier píxel de un subconjunto de píxeles conectados se denomina componente conectado, incluso si solo hay un componente conectado, el conjunto seguirá denominado conjunto conectado (GONZALEZ y WOODS, 2008).

Figura 6 –Representación de matriz de una imagen binaria,'8-adyacencia' del píxel central y 'adyacencia m' del píxel central respectivamente.

$$\begin{matrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \quad \begin{matrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \quad \begin{matrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Fuente: Totalmente generado por el autor,2020.

Llamarlo 'R' como un subconjunto de píxeles de una imagen. Donde este subconjunto es un conjunto conectado se llamará una región. Un ejemplo utilizado por González y Woods (2008), se considera dos regiones y se dice como adyacente, además de su unión forma un conjunto conectado.

Las regiones que no son adyacentes se denominan desarticuladas. Cuando se trata de regiones se considera la 'adyacencia 4" y '8-adyacencia' (Figura 7), donde dar sentido lógico se debe especificar el tipo de adyacencia que se utilizará. Como se ve en el cuadro 6, si se utiliza en '4-adyacencia' las dos regiones no están conectadas porque no hay trayecto válido.

Figura 7 -Dos regiones donde se utilizó la adyacencia 8 en relación con los píxeles 1.

$$\left.\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}\right\} R_i \qquad \left.\begin{matrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}\right\} R_j$$

Fuente: González y Woods, 2008.

Teniendo en cuenta una región donde prevalece la mayoría de los píxeles del mismo valor, que tienen una "conectividad 4", es posible observar la existencia de dos tipos de píxeles dependiendo de su ubicación. Donde los píxeles que tienen una "conectividad 4" completa se denominan la línea inferior o de fondo y los píxeles que no tienen esta característica se denominan la línea de frente (GONZALEZ y WOODS, 2008).

Figura 8 –Pixel de Linha de fundo ou plano de fundo

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Fuente: González y Woods, 2008.

Como se muestra en la Figura 8, se puede observar la ubicación de un borde, es decir, la delimitación de un área de píxeles comunes entre sí. Los píxeles que están en el borde de esta área son los píxeles del borde o también como se describió anteriormente son los píxeles de primera línea.

Para las mediciones de distancia de píxeles, descritas por Filho y Neto (1999) si los píxeles p, q y z se estipulan respectivamente en coordenadas (x, y), (s, t) y (u, v). La función de distancia D se puede establecer siguiendo las siguientes propiedades:

- $D(p,q) \geq 0$ (si y sólo si); $D(p,q) = 0 \quad p = q$
- $D(p,q) = D(q,p)$;
- $D(p,z) \leq D(p,q) + D(q,z)$;

Filho y Neto (1999) también mencionan las distancias euclidianas, las distancias D4 y las distancias D8. La distancia euclidiana se refiere al radio entre los dos puntos. La distancia D4 también conocida como City-Block es una forma centrada en píxeles de rand (x,y). La Distancia D8 llama al tablero de ajedrez generar la forma de un cuadrado centrado en el píxel

(x,y).

2.1.4 Herramientas utilizadas en el procesamiento de imágenes

Una herramienta importante en el procesamiento de imágenes es el filtrado, que se utiliza normalmente para el filtrado espacial y el filtrado de dominio de frecuencia. Donde el método espacial funciona con convoluciones de máscaras directamente en la matriz de píxeles, la imagen digital. El método de dominio de frecuencia, a su vez, utiliza la transformación de Fourier que se va a realizar (FILHO y NETO, 1999).

Según Russel y Norvig (2004), convertir las características de una imagen en un modelo de objetos conocidos es un proceso conocido como reconocimiento de objetos. Donde consta de tres pasos: segmentación de la imagen en objetos distintos, la posición y orientación de cada objeto en el punto de vista del observador y la forma especificada de cada objeto. El reconocimiento es el paso que clasifica los objetos encontrados, teniendo en cuenta un conjunto de conocimientos previamente establecidos.

2.2 Sudoku

Siendo uno de los juegos de números más populares desde el año 2005. el sudoku. que en la traducción literal es “número en su lugar”, para su resolución se utiliza combinaciones lógicas de prueba y error (CHEN et al, 2016).

2.2.2 Histórico

En 1780, Euler demostró la idea de organizar un conjunto particular de números para que cualquier número (o símbolo) se produzca una sola vez en cada fila o columna. Las Reglas del Sudoku han sido propuestas actualmente por Howard Garns, y la expansión de la restricción de Euler es válida para las regiones 3x3 (PAL et al, 2018). Un ejemplo de juego se muestra en la Figura 9.

Figura 9 – Un juego de Sudoku (a) y su solución(b).

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

a

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

b

Fuente: Autoral generado por el generador de sudoku,2020.

En 1984, Maki Kaji publicó el juego en la revista Nikoli una revista de rompecabezas, donde le dio al juego su nombre moderno de sudoku. La versión estándar del juego se reduce a una mesa de 9x9 que cuenta con 81 espacios y 3x3 regiones. Todo sudoku comienza con algunos espacios ya llenos, y teniendo el objetivo de llenar los 81 espacios siguiendo ciertas reglas (RAOUF et al, 2014).

2.2.3 Reglas del juego

Como afirman Deshmukh et al (2016), el sudoku es un rompecabezas combinatorio basado en la lógica 9X9. El rompecabezas proporciona una cuadrícula parcialmente completada, que normalmente tiene una única solución. Un Sudoku completamente lleno satisface la propiedad de que cada una de las nueve cuadrículas 3X3 de Sudoku contiene cada número del 1 al 9 exactamente una vez.
Para la resolución del sudoku es necesario seguir tres reglas según Raouf et al (2014), que son:

•En cada línea debe contener los números 1 a 9 de manera diferente, es decir, los números no deben repetirse en la misma línea, como se muestra en la figura 10:

Figura 10 -Ejemplo de fila resuelta.

3	5	1	7	2	4	8	9	6

Fuente: Totalmente generado por el autor, 2020.

•En cada columna debe contener los números del 1 al 9 de manera diferente, es decir, los números no deben repetirse en la misma línea, como se muestra en la Figura 11:

Figura 11 - Ejemplo de columna resuelto.

3
8
9
4
2
5
1
7
6

Fuente: Totalmente generado por el autor, 2020.

•En cada región 3x3 debe contener los números 1 a 9 de manera diferente, es decir, los números no deben repetirse en la misma línea, como se muestra en la figura 12:

Figura 12 - Ejemplo de región resuelta.

3	5	1
8	6	7
9	4	2

Fuente: Totalmente generado por el autor, 2020.

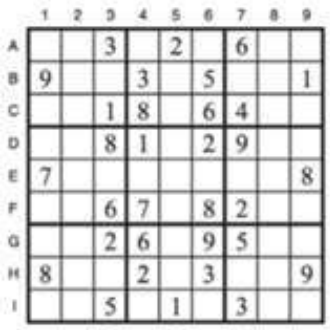
El juego es atractivo en la opinión de Godinho (2008), porque las reglas son simples, pero el razonamiento necesario para lograr la solución puede ser complejo. Godinho (2008) incluso dice que “Sudoku es recomendado por algunos educadores como un ejercicio para el pensamiento lógico. El nivel de dificultad se puede seleccionar de acuerdo con el público para el que está destinado.”

2.2.4 Algoritmo de solución

Chen et al (2016) describen que el sudoku puede considerarse un problema de satisfacción de las restricciones, es decir, un PSR. Sudoku es entonces un PSR con 81 variables y cada espacio de tabla ocupa una sola variable, y solo los dígitos 1 a 9 son válidos para cada variable.

Las variables se pueden nombrar siguiendo la siguiente definición, observando la primera línea de izquierda a derecha si tiene A1 a A9, mientras que cuando mira las columnas que ha seguido la dirección de arriba hacia abajo tenemos las variables de A1 a I1 como se muestra en la Figura 12.

Figura 12 - Ejemplo de la orientación de las variables como PSR.



	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

Fonte: CHEN et al, 2016.

Un algoritmo utilizado para resolver problemas de satisfacción de restricciones es el backtracking. El algoritmo selecciona el primer espacio vacío, le asigna un dígito y comprueba si hay conflictos. Si no tiene el proceso se repite en el segundo espacio, sin embargo, si el conflicto se produce se asignará otro dígito (CHEN et al, 2016).

La idea básica de backtracking es que si no hay ningún conflicto en el espacio actual asigna un valor al espacio siguiente, si no es posible vuelve al espacio anterior y le reasigna un dígito distinto. Este es el proceso conocido como retroceso (CHEN et al, 2016).

2.3 Trabalhos relacionados

Algunos algoritmos de solución para Sudoku han sido implementados por diferentes autores. A continuación, se presentaron brevemente diferentes soluciones.
Según Weyland (2015), cuando se utiliza la búsqueda armónica (HS), implica que su uso por sí mismo no ofrece ninguna novedad. El uso del algoritmo se pega al rendimiento que se puede obtener con una evolución de las estrategias y que incluso después de años de investigación no ofrecía resultados significativos.

Según Raouf et al (2014), con el método híbrido del algoritmo de polinización con la investigación de la armonía caótica (FPCHS). Raouf et al obtuvieron un resultado numérico más preciso y preciso en comparación con el uso di un método básico de búsqueda armónica.
El algoritmo híbrido cuántico-clásico fue implementado por Pal et al (2018) para la resolución sudoku de regiones 4x4 incompletas, que resultó ser un método exponencialmente más rápido.

Chen et al (2016) llevaron a cabo un estudio con backtracking algoritmos y algoritmos de enlace de baile (algoritmo x) donde resultó más lento debido a la falta de una base de datos anterior, pero resultó más eficiente en sudokus más complejos. En el Cuadro 1 se presenta un resumen de los métodos empleados por los autores antes mencionados:

Tabla 1 - Algoritmos utilizados por otros autores.

Autor	Año	Algoritmo utilizado	Observes
Weyland	2015	Hs	Resultó ineficiente
Raouf et al	2014	FPCHS	Es más preciso y preciso que el HS
Pal et al	2018	híbrido cuántico-clásico	Es más rápido en sudokus 4x4
Chen et al	2016	Algoritmo x	Es más eficiente en sudokus complejos

Fuente: Totalmente generado por el autor, 2020.

3. RESULTADOS

Al realizar investigaciones sobre los algoritmos de solución para el sudoku, se encontraron las soluciones más diversas con diferentes grados de complejidad para la resolución computacional de un juego tan famoso que es sudoku.
Como se ve en el trabajo de Felisberto (2016) donde hizo un trabajo detallado sobre la ejecución de la construcción de un robot solver de sudoku donde durante su artículo el mismo consideró el uso de algoritmos de solución como Backtracking, Dance Links y Ruled based. Cuando al final se decide el uso del retroceso con la ayuda de algunos dispositivos como las técnicas de valores mínimos restantes (MVR) y comprobación directa (FC).

Sin embargo, debido a que no hay profundidad en relación con el rendimiento de la solución no se puede comparar con los otros estudios para no presentar datos de comparación, pero es extremadamente detallado en su ejecución.

El trabajo de Kamal et al (2015) compara tres tipos de algoritmos que son Backtracking, Simulated Annealing y Genetic Algorithm. Cuando los resultados obtenidos de su trabajo apuntan a una mayor eficiencia del algoritmo de retroceso en relación con el otro, principalmente en relación con el sudoku de mayores grados de complejidad, donde el tiempo de resolución es considerablemente más corto en comparación con el algoritmo genético y ligeramente inferior al de recocido simulado simulado.

Al igual que en el trabajo de Kamal et al (2015) donde los otros algoritmos estudiados tuvieron un rendimiento inferior al retroceso, otro artículo que demuestra resultados de comparación es el trabajo de Berggren y Nilsson (2012) donde el retroceso fue el segundo algoritmo más eficiente. sin embargo, no fue capaz de resolver aproximadamente 2.3% de los puzzles probados. mientras que los otros algoritmos

probados como el gobernado base fue el algoritmo probado más rápido además de haber logrado resolver todos los puzzles. El tercer algoritmo visto fue el solver de la máquina boltzmann que no funcionaba tan bien como los otros donde tenía que tener 46 pistas para tener ejecuciones razonables siendo que era incapaz de resolver el 2,5% de los puzzles.

El estudio de Job y Paul (2016) se llevó a cabo con el objetivo del algoritmo de retroceso, donde se probaron 30 rompecabezas divididos en el número de pistas, de 17 a 22 pistas. El resultado fue un tiempo medio de resolución del rompecabezas de 20.365 segundos, con los puzzles con más pistas teniendo un tiempo más corto en comparación con los puzles de menos pistas de 3.24 segundos. Otro trabajo que también tuvo resultados similares fue el artículo de Majumder et al (2010), y la plataforma utilizada para resolver puzzles fue en entorno java donde tenía resultados en comparación con otro software un tiempo de resolución de 20 milisegundos.

Schottlander (2014) llevó a cabo un experimento en el que utilizó dos enfoques distintos utilizando el algoritmo de retroceso, que son opciones ordenadas y desordenadas. Incluso si el enfoque tiene el potencial de resolver los puzzles más rápidamente se observó lo contrario, especialmente en juegos con menos pistas, sin embargo en juegos con mayor número de pistas el rendimiento era prácticamente el mismo. Un trabajo que realizó pruebas de comparación entre algoritmos de fuerza bruta, Deliberación simulada. Algoritmo Genético, Búsqueda de Armonía y Referencia de Gráficos fue realizada por Chatterjee et al (2014). Se obtuvo como resultado que el algoritmo de mayor velocidad era la referencia de gráficos, haciendo una fuerza bruta bien emparejada, con algunos casos siendo más rápido. También se notó que la búsqueda de Harmony tenía la peor velocidad llegando a los casos de 100 segundos para resolver un rompecabezas.

El artículo de Bunma y Chongstitvatana (2010) muestra que el algoritmo genético tiene una buena tasa de éxito, y en su estudio sólo no pudo resolver los problemas considerados de nivel extremo. Mientras que los otros niveles de dificultad tenían una tasa de solución del 100%.

Chi y Lange (2012) realizaron estudios de los algoritmos de retroceso y recocido simulado, así como de proyecciones alternativas. Es notable que el rendimiento más eficiente y rápido fue el de retroceso en rompecabezas aplicados donde la tasa de éxito fue de 1.0 y la resolución media de 0.008 segundos con casos de rompecabezas más complicados que alcanzan hasta el tiempo de 0.031 segundos para la resolución. Mientras que en estos mismos casos simulado recocido alcanzó 7.2 segundos y Proyecciones Alternativas 0.5 segundos. y ambos no tenían una alta tasa de resolución siendo un promedio de 0,75 y 0,72 respectivamente.

Se ve que el algoritmo de retroceso es más rápido que la Fuerza Bruta según Eremic et al (2013). Donde esa comparación se realizó usando rompecabezas con diferentes cantidades de pistas siendo 36 para fácil, 32 para el medio y 28 para duro. El algoritmo de retroceso completó los puzzles a 6 ms, 42 ms y 46 ms, mientras que la fuerza bruta tomó más de 90 minutos en todas las dificultades.

En 2015 Sabuncu desarrolló un algoritmo para resolver rompecabezas sudoku que fue llamado el algoritmo híbrido de optimización de colonias de hormigas. Este algoritmo desarrollado es un híbrido de algoritmo de colonia de hormigas y métodos analíticos de solución sudoku. Sus pruebas en su algoritmo de rompecabezas considerado difícil fue que es capaz de resolverlo en menos de 2 segundos.

Wang et al(2015) en su trabajo utilizaron el Algoritmo Genético con una diferencia porque aplicaron un método de Mutaciones Filtradas donde el uso de las reglas de rompecabezas que se aplican después de que los candidatos elegibles de cada celda son elegidos por el algoritmo convencional aplicando así el número más cohesivo a la solución en lugar de elegir aleatoriamente entre los elegibles. Con este método logró una mejora en la tasa de éxito del algoritmo de 4.167% a 65.52% en rompecabezas de mayor dificultad.

El algoritmo aplicado por Zhai y Zhang (2013) conocido como Entropía de Información Personalizada donde sus resultados fueron muy positivos. En su trabajo se hace la comparación con el algoritmo genético donde tanto en la tasa de éxito y la velocidad el algoritmo de la entropía de información fue más eficaz principalmente en rompecabezas más complejos y el tiempo medio de resolución es de 5,3 ms, mientras que el algoritmo genético pasó 14 segundos con una tasa de éxito de 0,9 y 0,7.

Borges et al (2016) aplicaron el método de tinción de gráficos para la resolución de sudoku. Para su artículo desarrollaron dos algoritmos donde el algoritmo llamado como algorithm2 tuvo el mejor rendimiento siendo la resolución de problemas de sudoku 9x9 en menos de 0.02 segundos, pero también realizó pruebas en rompecabezas con cuadrículas más grandes como 25x25 donde tenía un tiempo de resolución de 46.33 segundos.

En 2009, Pacurib et al llevaron a cabo el estudio de la solución con el algoritmo llamado Colonia De abeja artificial, sus resultados no fueron muy expresivos en relación con la velocidad de resolución de los puzzles donde aproximadamente 4, 12, 176 segundos se gastaron en puzzles considerados fáciles, medios y difíciles respectivamente.

Almog (2009) llevó a cabo un estudio comparando el rendimiento de tres algoritmos que son: Quantum Simulated Annealing (QSA), Algoritmo Genético Cultural y un algoritmo híbrido entre ellos, el QSA fue el que obtuvo el mejor éxito y tasa de rendimiento resolviendo el 64% de los puzzles como un promedio de 40 segundos.

Gunther y Moon (2014) utilizaron un algoritmo de minimización de Entropía en 2012 para resolver más de 4000 rompecabezas de sudoku donde se dividieron por igual en fáciles, medios, difíciles y malvados. El algoritmo tuvo buenos resultados y la estrategia fue nombrada Como C, obtuvo rendimientos en cada nivel respectivamente de 0.1; 0,13; 0.22 y 0.29 segundos siendo incapaz de resolver sólo 2 rompecabezas de nivel maligno.

Mantere en 2008 probó el uso de Algoritmo Cultural para la solución sudoku. Sus resultados muestran que dicho algoritmo presenta un rendimiento ligeramente más eficiente en comparación con el algoritmo genético.
En 2006 Mantere llevó a cabo un trabajo utilizando algoritmo genético, donde dice que el algoritmo genético de una manera pura, es decir, sin ninguna regla específica del problema para el sudoku no es muy eficiente en la resolución de los puzzles, pero son solucionables.

El estudio de Sato e Inoue (2010) utilizó algoritmo genético y realizó algunos cambios, como el uso de una función de búsqueda local entre otros métodos que resultaron en una mejora significativa en el rendimiento del algoritmo. Lim et al (2015) también realizaron la prueba de comparación de su algoritmo genético de par monógamo con el algoritmo Sato e Inoue donde se realizó de manera similar y una diferencia

del 1% en la tasa de éxito.

Soto et al(2014) realizaron un estudio de comparación entre los algoritmos AC3 en dos métodos, siendo Tabu Search y Cuckoo Search with the Genetic Algorithm, donde demostraron que los métodos AC3 tenían el uso del 100% en la resolución de rompecabezas con la limitación de sólo cien mil interacciones mientras que el Algoritmo Genético tenía sólo un 6.6% de utilización de los puzzles con esta limitación. También se realizó sin el límite de interacciones y todos los algoritmos tenían una utilización del 100%.

En 2015, Soto et al demostraron en un artículo el funcionamiento del algoritmo AC3 con la adición de técnicas de filtrado, que resultaron ser muy útiles para facilitar el proceso de resolución del rompecabezas. También en 2015 Soto et al realizaron una comparación del algoritmo basado en el teorema del matrimonio con AC3 de su estudio anterior, donde se observó que obtuvo un rendimiento similar.

Harrysson y Laestander (2014) escribieron que el algoritmo Dance Links resuelve el rompecabezas de manera eficiente. El algoritmo basado en Ant Colony Optimization propuesto por Lloyd y Amos (2019), donde el algoritmo demuestra un rendimiento más eficiente en problemas más difíciles que algoritmos como Backtracking y Dance Links tanto en tiempo de resolución como en tasa de éxito;

La conclusión que Weiss (2009) tuvo en la realización de su trabajo en algoritmo genético para la solución Sudoku, es que el algoritmo realiza la función mal en comparación con backtracking, al menos en rompecabezas de cuadrícula 9x9. Como a veces es necesario que el proceso se reinicie para lograr el resultado, porque ocurre que el algoritmo se atasca, incapaz de alcanzar la solución, que se presenta para ser un problema de optimización para el uso que se impone.

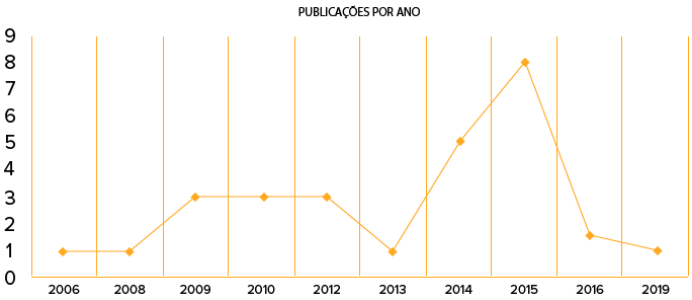
Lindberg y Steier (2015) propusieron el uso de un algoritmo híbrido de búsqueda AC3 y Taboo. Los autores describieron que tal algoritmo no es apropiado para la resolución del rompecabezas, porque en comparación con el método Fuerza bruta el algoritmo realizado más bajo, ya sea la tasa de éxito o el tiempo de resolución del rompecabezas. También describen que no creen que la optimización del algoritmo sea posible.

Coelho y Laporte (2014) expresan en su artículo que el algoritmo de retroceso tiene una mejora de la eficiencia cuando se basa en criterios heurísticos. También demuestra que el algoritmo Backtracking with Heuristics presenta una constancia en el tiempo de resolución independientemente del número de pistas, con una ligera variación en los tiempos de solución.

Borges et al (2016) aplicaron el método de tinción de gráficos para la resolución de sudoku. Para su artículo desarrollaron dos algoritmos donde el algoritmo llamado como algorithm2 tuvo el mejor rendimiento siendo la resolución de problemas de sudoku 9x9 en menos de 0.02 segundos, pero también realizó pruebas en rompecabezas con cuadrículas más grandes como 25x25 donde tenía un tiempo de resolución de 46.33 segundos.

Después de leer los artículos fue posible recopilar cierta información, es posible notar que las publicaciones vistas comenzaron en 2006.é visible que en 2015 tuvo un aumento en el número de publicaciones relacionadas con el tema como se ve enel gráfico 1.

Figura 1 - Publicaciones relacionadas con el tema cada año.



Fuente: Totalmente generado por el autor, 2020.

La Tabla 2 muestra los algoritmos más frecuentes en las publicaciones, y desde 2010, la mayoría de los artículos vistos utilizaban retroceso para sus estudios.

Tabla 2 - Publicaciones y algoritmos más cubiertos por el período.

Período	Quiantidade	Técnica mais frequente
2004-2010	5	Algoritmo genético
2010-2015	12	Retroceso
2015-corriente	11	Retroceso

Fuente: Totalmente generado por el autor, 2020.

En este período tuvo varios estudios utilizando los enfoques más diferentes. Se demuestra en la tabla que los algoritmos más abordados fueron los algoritmos de retroceso, algoritmo genético y el recocido simulado.

Tabla 3 - Número de veces que el algoritmo fue abordado o citado.

Técnica utilizada	Cantidad
Retroceso	13
Algoritmo genético	11
Anconte simulado	4

Fuente: Totalmente generado por el autor, 2020.

A partir de los estudios que proporcionaron datos de comparación de los algoritmos mencionados anteriormente, como el tiempo de ejecución y la tasa de éxito, disponibles en la Figura 13. Sin embargo, no es posible hacer una comparación directa entre los métodos aplicados, porque los estudios se realizaron en diferentes plataformas y máquinas.

Figura 13 - Datos sobre la eficiencia y la tasa de éxito de los algoritmos.

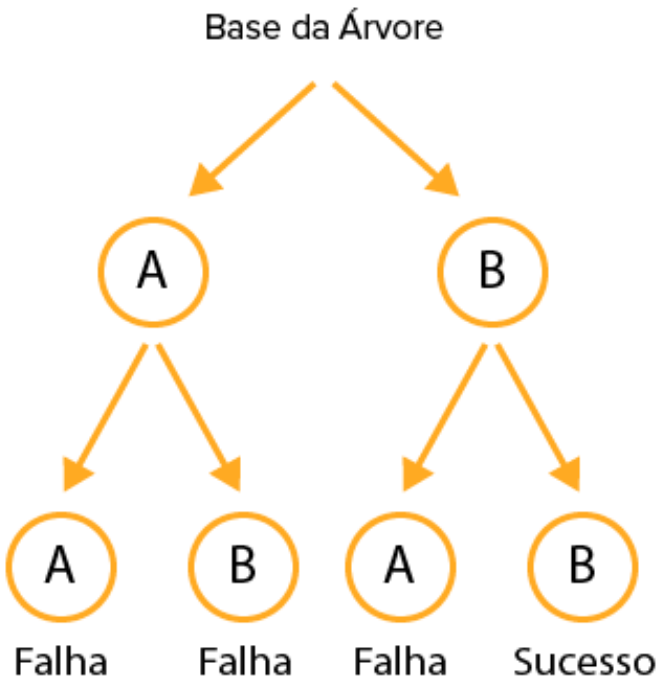
	Almong (2009)		Chl e Lange (2012)		Chatterje et al (2014)		Kamal et al (2015)	
	Taxa de sucesso	Tiempo de execução (s)	Taxa de sucesso	Tiempo de execução (s)	Taxa de sucesso	Tiempo de execução (s)	Taxa de sucesso	Tiempo de execução Log (s)
Backtracking / Brute force	xxx	xxx	1	0,008	xxx	0,051733	xxx	0,1
Genetic Algorithm	0,13	75,75	xxx	xxx	xxx	3,883	xxx	5000
Simulated Annealing	0,64	40,2	0,97	0,753	xxx	0,526667	xxx	5
Obs:	xxx		Dados: médida do nivel Hard		Taxa de sucesso definida como 1		Dados: Puzzle com 25 dicas	

Fuente: Totalmente generado por el autor, 2020.

Se puede tomar de la figura 13 mostrada anteriormente, que el algoritmo de retroceso funciona bien como una buena tasa de éxito. Es notable que de los algoritmos más abordados tiene un rendimiento ligeramente superior en comparación con el Annealing Simulado y ambos tienen una gran disparidad en comparación con el Algoritmo Genético.

El algoritmo de retroceso, aunque es un algoritmo de baja complejidad en relación con el otro, funciona bien. El retroceso se comporta como se expresa en la Figura 14.

Figura 14 - Ejemplo de la operación de retroceso.



Fuente: Totalmente generado por el autor, 2020.

La lógica de funcionamiento de retroceso es bastante simple, donde el algoritmo establece una ruta de operación que seguirá a las decisiones tomadas hasta que se produzca un error o conflicto con las reglas estipuladas. Si se produce este error, el algoritmo devuelve un paso anterior de la causa del conflicto e intenta una nueva rama del árbol de decisión, siguiendo así la resolución que se considera adecuada. Como se muestra en la figura 14.

4. DISCUSIÓN Y CONCLUSIONES

Como se ve en este artículo, donde se demostró la parte teórica sobre el procesamiento de imágenes y cómo esta técnica se puede aplicar al escaneo de rompecabezas Sudoku. Se presentan varios algoritmos, que complementan la técnica de procesamiento de imágenes con el fin de obtener el resultado del rompecabezas deseado.

Varios de los algoritmos vistos en este artículo tienen una gran tasa de éxito, así como un tiempo de ejecución bajo. Entre los algoritmos que se ajustan a esta descripción, el algoritmo BackTracking fue resaltado por su baja complejidad y a su vez fácil comprensión, donde no tiene una tasa de éxito perfecta. es ideal para aplicaciones y modificaciones de la misma, como se ve en el trabajo de Felisberto (2016). Lo que construyó un robot solver Sudoku que no requería algoritmos más complejos para lograr el resultado esperado.
Sudoku es un hermoso desafío matemático, que permite su uso para pruebas y desarrollo de algoritmos capaces de resolverlo con éxito. Un algoritmo que mostró avances significativos fue el algoritmo genético a partir de nuevos enfoques como Sato e Inoue (2010).

Un contratiempo observado en el estudio bibliográfico fue la falta de estandarización de los puzzles utilizados en las pruebas. Donde en cada publicación a menudo utilizaban diferentes rompecabezas y también los clasificaban por diferentes niveles de dificultad. Esta práctica impide una comparación más objetiva de las interpretaciones obtenidas en cada publicación. Dado que para un desarrollo más claro y comparaciones más objetivas del trabajo futuro se recomienda estandarizar los puzzles utilizados en las pruebas o simplemente la adición de ellos en apéndices para que los futuros investigadores puedan utilizar los mismos puzzles.

REFERENCIAS

- A. Deshmukh, A. Nagraj, M. A. Radke, (2016). SudoKrypt: A Novel Sudoku based Symmetric Encryption Scheme. Journal of Network and Information Security Volume, 4(2).
- A. Majumder, A. Kumar, N. Das, N. Chakraborty, (2010). The game of Sudoku-advanced backtrack approach. IJCNS–International Journal of Computer Science and Network Security, 10, 255-258.
- A. Pal, S. Chandra, V. Mongia, B. K. Behera, P. K. Panigrahi, (2018). Solving Sudoku Game Using Quantum Computation.
- C. S. G. Godinho, (2008). Alguma da matemática do sudoku (Doctoral dissertation, Universidade de Aveiro).
- C. Solomon, T. Breckon, (2011). Fundamentals of digital image processing: a practical approach with examples in Matlab. Wiley-Blackwell.
- D. Bunma, P. Chongstitvatana, (2010). Solving Logic Puzzles with Genetic Algorithm.
- D. Job, V. Paul, (2016). Recursive Backtracking for Solving 9* 9 Sudoku Puzzle. Bonfring International Journal of Data Mining, 6(1), 07-09.
- E. C. Chi, & K. Lange, (2012). Techniques for solving sudoku puzzles. arXiv preprint arXiv:1203.2295.
- G. Zhai, J. Zhang, (2013). Solving sudoku puzzles based on customized information entropy. International Journal of Hybrid Information Technology, 6(1), 77-91.
- H. Lloyd, M. Amos, (2019). Solving Sudoku with Ant Colony Optimization. IEEE Transactions on Games.
- I. Sabuncu, (2015, March). Work-in-progress: solving Sudoku puzzles using hybrid ant colony optimization algorithm. In 2015 1st International Conference on Industrial Networks and Intelligent Systems (INISCom) (pp. 181-184). IEEE.
- J. A. Pacurib, G. M. M. Seno, J. P. T. Yusiong, (2009, December). Solving sudoku puzzles using improved artificial bee colony algorithm. In 2009 fourth international conference on innovative computing, information and control (ICICIC) (pp. 885-888). IEEE.
- J. Almog, (2009). Evolutionary computing methodologies for constrained parameter, combinatorial optimization: Solving the Sudoku puzzle. Africon 2009. https://doi.org/10.1109/afrcon.2009.5308284
- J. E. R. Queiroz, & H. M. Gomes, (2006). Introdução ao processamento digital de imagens. Rita, 13(2), 11-42.
- J. Gunther, T. Moon, (2012). Entropy Minimization for Solving Sudoku. IEEE Transactions on Signal Processing, 60(1), 508–513. https://doi.org/10.1109/tsp.2011.2169253
- J. Lindberg, M. Steier, (2015). Efficiency of the hybrid AC3-tabu search algorithm for solving Sudoku puzzles.
- J. M. Weiss, (2009, April). Genetic algorithms and sudoku. In Midwest Instruction and Computing Symposium (MICS 2009) (pp. 1-9).
- L. C. Coelho, G. Laporte, (2014). A comparison of several enumerative algorithms for Sudoku. Journal of the Operational Research Society, 65(10), 1602–1610. https://doi.org/10.1057/jors.2013.114
- L. Chen, Y. Guo, S. Wang, S. Xiao, K. Kask, (2016). Sudoku Solver.
- L. P. Specht, O. Khatchatourian, R. T. Dos Santos, (2013). Measurement of pavement macrotexture through digital image processing. Acta Scientiarum. Technology, 35(1), 31-38.

- M. Eremic, R. Adamov, N. Bilinac, V. Ognjenovic, V. Brtka, I. Berkovic, (2013). COMPARISON OF STATE SPACE SEARCH ALGORITHMS-SUDOKU PUZZLE GAME. Chief and responsible editor, 154.
- M. Harrysson, H. Laestander, (2014). Solving Sudoku efficiently with Dancing Links.
- M. Schottlender, (2014). The effect of guess choices on the efficiency of a backtracking algorithm in a Sudoku solver. IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014. https://doi.org/10.1109/lisat.2014.6845190
- O. A. Raouf, I. E. Henawy, M. A. Baset, (2014). A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles. International Journal of Modern Education and Computer Science, 6(3), 38–44. https://doi.org/10.5815/ijmecs.2014.03.05
- O. M. Filho, H. V. Neto, (1999). Processamento digital de imagens. BRASPORT.
- P. Berggren, D. Nilsson, (2012). A study of Sudoku solving algorithms. Royal Institute of Technology, Stockholm.
- R. C. Gonzalez, R. E. Woods, (2002). Digital image processing. Prentice Hall.
- R. C. Gonzalez, R. E. Woods, (2008). Digital image processing. Parson.
- R. C. Gonzalez, R. E. Woods, (2018). Digital image processing. Pearson.
- R. C. Gonzalez, R. E. Woods, S. L. Eddins, (2004). Digital Signal Processing Using Matlab. Pearson/Prentice Hall.
- R. Soto, B. Crawford, C. Galleguillos, F. C. Venegasy, F. Paredes, (2015).
- A Marriage Theorem Based-Algorithm for Solving Sudoku. 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI). https://doi.org/10.1109/micai.2015.24
- R. Soto, B. Crawford, C. Galleguillos, K. Crawford, F. Paredes, (2015).
- A Filtering Technique for Helping to Solve Sudoku Problems. Communications in Computer and Information Science HCI International 2015 - Posters' Extended Abstracts, 598–603. https://doi.org/10.1007/978-3-319-21380-4_102
- R. Soto, B. Crawford, C. Galleguillos, S. Misra, E. Olguin, (2014). Solving sudokus via metaheuristics and AC3. 2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST). https://doi.org/10.1109/icastech.2014.7068127
- S. Borges, T. Lima, V. Marques, (2016). Coloração de Grafos Aplicado na resolução do Sudoku.
- S. Chatterjee, S. Paladhi, R. Chakraborty, (2014). A Comparative Study On The Performance Characteristics Of Sudoku Solving Algorithms. IOSR Journal of Computer Engineering, 16(5), 69–77. https://doi.org/10.9790/0661-16566977
- S. J. Russell, P. Norvig, (2004). Artificial intelligence: a modern approach. Prentice Hall/Pearson Education
- S. Kamal, S. S. Chawla, N. Goel, (2015). Detection of Sudoku puzzle using image processing and solving by Backtracking, Simulated Annealing and Genetic Algorithms: A comparative analysis.
- 2015 Third International Conference on Image Information Processing (ICIIP). https://doi.org/10.1109/iciip.2015.7414762
- T. Mantere, J. Koljonen, (2006, October). Solving and rating sudoku puzzles with genetic algorithms. In New Developments in Artificial Intelligence and the Semantic Web, Proceedings of the 12th Finnish Artificial Intelligence Conference STeP (pp. 86-92).
- T. Mantere, J. Koljonen, (2008). Solving and analyzing Sudokus with cultural algorithms. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). https://doi.org/10.1109/cec.2008.4631350
- T. Y. Lim, M. A. Al-Betar, A. T. Khader, (2015). Monogamous pair bonding in genetic algorithm. 2015 IEEE Congress on Evolutionary Computation (CEC). https://doi.org/10.1109/cec.2015.7256869
- T. Z. Felisberto, (2016). Robô Solucionador de Sudoku.
- Y. Sato, H. Inoue, (2010). Solving Sudoku with genetic operations that preserve building blocks. Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games. https://doi.org/10.1109/itw.2010.5593375
- Z. Wang, T. Yasuda, K. Ohkura, (2015). An evolutionary approach to sudoku puzzles with filtered mutations. 2015 IEEE Congress on Evolutionary Computation (CEC). https://doi.org/10.1109/cec.2015.7257096