

# Un algoritmo sencillo de Sudokus de $9 \times 9$ \*

Armando Cabrera Pacheco,  
Edwin Meneses Rodríguez  
y  
Roger Pacheco Castro

Facultad de Matemáticas  
Universidad Autónoma de Yucatán

kabrez@gmail.com, edwin.meneses.r@gmail.com, matbnt@gmail.com

RECIBIDO: 15 de junio de 2009

ACEPTADO: 2 de julio de 2009

## RESUMEN

El objetivo de este trabajo fue diseñar un algoritmo sencillo para generar Sudokus iniciales de  $9 \times 9$  con solución única con distintos niveles de dificultad. Este trabajo revisa la literatura para conocer las fronteras del conocimiento sobre este tópico de investigación. Se halló una lista de Sudokus iniciales de orden  $m = 64$  con solución única, dicho orden es el máximo encontrado hasta la fecha. A partir de los elementos de esta lista, los cuales fueron llamados elementos básicos, se creó un algoritmo sencillo de entender e implementar para generar Sudokus iniciales con solución única.

## 1. INTRODUCCIÓN

El Sudoku es un tipo especial de Cuadro latino, y abundaremos en detalles más adelante. Este último apareció en la literatura árabe, hace aproximadamente 700 años (véase la página [The Internet Encyclopedia of Science](#) en la sección [GAMES & PUZZLES](#)). Como rompecabezas el Cuadro latino de  $n \times n$  con-

---

\*Este trabajo es la traducción al español del original desarrollado en el marco del concurso *Mathematical Contest in Modeling 2008* realizado por el COMAP del 14 al 18 de febrero de 2008, obteniendo el resultado de "Mención Honorífica".

siste en asignar  $n$  símbolos distintos (generalmente los números enteros del 1 al  $n$ ) a una matriz cuadrada de tamaño  $n \times n$  de modo que cada uno de los  $n$  símbolos ocupe exactamente una celda en cada renglón y cada columna. Esta clase de “rompecabezas” fue redescubierta por Euler [1707–1783] un siglo después, durante sus investigaciones sobre objetos combinatorios.

El primer Sudoku que aparece en la literatura fue publicado en 1974 por el arquitecto francés Howard Garns (“Number Place”, *Dell Magazine*, EUA, 1974). Este rompecabezas obtuvo gran popularidad en Japón en 1986, pero no fue hasta el año 2004 cuando logró una popularidad a nivel mundial. En los Estados Unidos de América, Garns lo denominó *Number place*, por otra parte, en Japón se dio a conocer como *Suuji wa dokushin*, que se puede traducir como: “Los números deben estar solos”; finalmente, desde 1997, se le conoce con el nombre de Sudoku, como abreviación de su nombre japonés (véase <http://www.freesudokupuzzle.org>).

La presentación más conocida del Sudoku es una matriz cuadrada de tamaño  $9 \times 9$ . Sin embargo, existe una generalización natural a una matriz de tamaño  $n^2 \times n^2$ , donde  $n$  es un número natural (Frank, 2005). Existen algunas variantes del Sudoku, como por ejemplo: el Sudoku samurai, el Sudoku flor, el Sudoku mariposa, el Sudoku asesino, etcétera (véase [www.cross-plus-a.com/sudoku.htm](http://www.cross-plus-a.com/sudoku.htm)). Nos restringiremos a considerar el Sudoku común.

Generalmente para construir un Sudoku de  $9 \times 9$  se consideran los números naturales del 1 al 9, sin embargo es posible construirlos utilizando nueve símbolos distintos. Esta construcción puede ser llevada por medio de ordenadores a partir de “generadores”; otra forma para construir estos rompecabezas es la japonesa, que es hecha a mano debido a que se piensa que un Sudoku hecho a mano es más entretenido que el generado por algún algoritmo (véase la página NIKOLI Co., Ltd.).

La razón principal por la cual muchas personas son atraídas por el Sudoku es que a pesar de que las reglas son simples, el razonamiento para lograr el camino correcto hacia la solución puede ser muy difícil, como veremos en los siguientes párrafos.

## 2. TEORÍA MATEMÁTICA DEL SUDOKU

A continuación se definirán algunos conceptos útiles para entender e implementar el algoritmo.

### 2.1. Definiciones básicas

La siguiente definición fue tomada de Springer Online Reference Works:

**Definición 2.1.** *Un Cuadro latino es una matriz cuadrada de  $n \times n$  en la cual cada fila y cada columna son permutaciones de los elementos de  $S = \{1, 2, \dots, n\}$  para  $n > 1$ .*

Es posible definir más de una relación de equivalencia sobre el conjunto de Cuadros latinos para una  $n$  fija. Una de ellas es la siguiente: decimos que dos Cuadros latinos son equivalentes si uno de ellos puede ser obtenido permutando las filas o columnas, o renombrando las entradas del otro.

A continuación definiremos lo que se entenderá en este trabajo por Sudoku, que ciertamente, es un subconjunto del conjunto de Cuadros latinos de  $9 \times 9$ .

**Definición 2.2.** *Un Sudoku es una matriz cuadrada de  $9 \times 9$  subdividida en nueve submatrices de  $3 \times 3$ , también denominadas “cajas” (como se muestra en la figura 1), en cuyas celdas se encuentran números del conjunto  $A = \{1, 2, \dots, 9\}$ , de tal forma que cada fila, cada columna y cada caja contiene uno y sólo uno de los elementos del conjunto  $A$ .*

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

FIGURA 1. Sudoku.

De manera similar a lo ocurrido con los Cuadros latinos y debido a que los Sudokus son un caso especial de éstos, es posible definir una relación de equivalencia sobre los Sudokus.

**Definición 2.3.** *Decimos que dos Sudokus son equivalentes si uno de ellos puede ser transformado en otro por medio de composiciones de las siguientes transformaciones:*

*Permutación de los dígitos  $1, 2, \dots, 9$ .*

*Reflexión.*

*Rotación ( $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  y  $270^\circ$ ).*

*Permutación de las columnas  $1$  a  $3$ ,  $4$  a  $6$  ó  $7$  a  $9$ .*

*Permutación de las filas  $1$  a  $3$ ,  $4$  a  $6$  ó  $7$  a  $9$ .*

**Definición 2.4.** *Un Sudoku inicial se obtiene al eliminar  $m$  números en las celdas de un Sudoku. El número  $m$  es el orden del Sudoku inicial ( $1 \leq m \leq 81$ ). Decimos que un Sudoku inicial está bien definido si y sólo si existe una única forma de llenar las celdas vacías para alcanzar un Sudoku.*

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

FIGURA 2. Sudoku inicial de orden  $m = 32$ .

## 2.2. Generación de Sudokus iniciales

Hallar en la literatura algoritmos que resuelven Sudokus iniciales no es complicado. De hecho, existen varios algoritmos útiles para resolverlos. El reto es crear un algoritmo que genere Sudokus iniciales bien definidos. Parece no existir, en la literatura, un método simple para determinar cuando un Sudoku inicial está bien definido sin necesidad de resolverlo, lo cual complica la creación de algoritmos para generar Sudokus iniciales con solución única.

En la literatura se encuentran algunos resultados que podrían ser útiles para generar Sudokus iniciales bien definidos. Por ejemplo, usando geometría algebraica, un Sudoku inicial es transformado en un problema de polinomios cuya solución se alcanza utilizando bases de Gröbner. Este método puede determinar si el Sudoku inicial tiene solución, y en caso afirmativo devuelve una solución (Gago-Vargas y cols., *Sudokus and Gröbner Bases: Not Only a Divertimento*).

Es evidente que el problema que se plantea en los párrafos anteriores es un reto y bien puede involucrar matemáticas avanzadas.

El objetivo es generar un algoritmo *sencillo* para generar Sudokus iniciales bien definidos con diferentes niveles de dificultad.

## 2.3. Algoritmos para generar y resolver Sudokus iniciales

En la literatura existen muchos algoritmos para generar y resolver Sudokus Iniciales. A continuación, se expone una breve explicación de algunos de ellos.

### ■ Algoritmos para resolver Sudokus iniciales:

- *Marcha atrás.* Consiste en asignar un valor aleatorio a una celda vacía, verificando que dicha asignación cumpla con las condiciones del Sudoku, de lo contrario, devuelve el Sudoku inicial original e intenta con un valor distinto. Este algoritmo itera este método conservando los valores que cumplen las condiciones del Sudoku. Es costoso en términos de tiempo (véase Wikipedia).
- *Programación lineal.* Transforma el Sudoku inicial en un problema de programación, con una función maximizadora (Bartlett y Langville, An Integer Programming Model for the Sudoku Problem).

### ■ Algoritmos para generar Sudokus iniciales:

- *Fuerza bruta.* Genera una matriz de  $9 \times 9$  y asigna los símbolos del conjunto  $\{1, 2, \dots, 9\}$  aleatoriamente, después comprueba si la matriz resultante es un Sudoku. En el momento en que encuentra un Sudoku, procede a eliminar el número de celdas requeridas por el usuario, verificando después de cada eliminación si la matriz resultante es un Sudoku inicial bien definido (Bartlett y Langville, An Integer Programming Model for the Sudoku Problem).
- *Método simétrico.* Para un Sudoku predefinido elimina celdas en la matriz triangular superior. Cuando las celdas están completamente determinadas las refleja por la diagonal y elimina las correspondientes celdas en la matriz triangular inferior para obtener un Sudoku inicial (Lee, 2006.).

Estos algoritmos pueden llegar a ser muy lentos (Marcha atrás y Fuerza bruta) y complicados (Programación lineal).

El objetivo de este trabajo recae en diseñar un algoritmo sencillo y fácil de entender que permita generar Sudokus con diferentes niveles de dificultad. Partiremos de una colección de Sudokus iniciales bien definidos, que según la literatura son aceptados como de “orden máximo”, y por medio de un algoritmo de solución, que se detallará más adelante, se obtendrá un Sudoku inicial bien definido con el nivel de dificultad deseado.

## 3. ALGORITMO

### 3.1. Objetivo

Generar, de forma sencilla, Sudokus iniciales bien definidos de orden  $m$  ( $0 < m < 64$ ) con diferentes niveles de dificultad.

### 3.2. Análisis del Problema

Considerando la existencia de algoritmos y la dificultad para determinar si el Sudoku inicial dado está bien definido, se encontró que puede resultar muy complicado crear un algoritmo sencillo (Hayet, Complejidad de los problemas, y Mancini, Sudoku game theory, models and algorithms).

Es evidente que las transformaciones entre los Sudokus iniciales bien definidos preservan la buena definición de éstos. Por lo tanto, de un solo Sudoku inicial bien definido, podemos obtener un gran número de Sudokus iniciales bien definidos (de hecho, solamente con las permutaciones obtenemos  $9!$  por cada Sudoku inicial), de tal forma que si no supiésemos que son equivalentes, sería muy difícil percatarse cuando dos Sudokus son equivalentes a simple vista. Entonces, de una pequeña lista de Sudokus iniciales bien definidos, podemos obtener una gran cantidad de Sudokus iniciales bien definidos.

Debido al gran interés que este juego ha generado en la gente, se ha intentado determinar cuál es el máximo orden de un Sudoku inicial bien definido. Aunque este misterio no se ha resuelto, existe una lista de 47621 rompecabezas de este tipo de orden  $m = 64$  (véase Minimum Sudoku), la cual es de gran utilidad para desarrollar un algoritmo que genere Sudokus iniciales bien definidos. Partiremos de esta lista para generar los Sudokus iniciales bien definidos, considerando que,  $m = 64$  es el orden máximo que se admite.

### 3.3. Procedimiento General

Con base en la lista de Sudokus iniciales bien definidos de orden  $m = 64$ , se procedió de la siguiente manera:

1. Se creó una pequeña base de datos con elementos de dicha lista, llamamos a estos elementos, *elementos básicos*.
2. El algoritmo toma aleatoriamente un elemento básico y lo modifica para obtener un Sudoku inicial bien definido equivalente con el nivel de dificultad deseado.

Para alcanzar esto, se encontró un algoritmo desarrollado en *Matlab*, el cual genera Sudokus iniciales simétricos y además proporciona una solución paso a paso del rompecabezas (Fasino, A "light" Sudoku Generator and Helper).

Del análisis del algoritmo, se concluyó que la función ("hint") que determina la solución paso a paso no es complicada, de esta forma podría ser útil para nuestro proyecto. La función básicamente consiste en dos estrategias:

- La primera estrategia intenta responder la pregunta "¿Qué número puede ocupar esta posición?".

- La segunda estrategia es preguntar “¿Dónde puede ir el número  $N$  en esta fila (o columna, o caja)?”.

Si alguna de las preguntas anteriores tiene solución única, el algoritmo asigna el valor correspondiente.

Este trabajo consistió en seleccionar un elemento básico de la base de datos, transformarlo y con ayuda de la función anterior completar el Sudoku inicial bien definido hasta el punto deseado. Esto último resulta de utilidad para determinar el grado de dificultad. Como resultado, se obtiene un algoritmo fácil y rápido que garantiza la unicidad de la solución.

### 3.4. El Algoritmo

Como un primer paso, se definen los niveles de dificultad. Se considera que los elementos básicos son los Sudokus iniciales bien definidos aceptados en la literatura como orden máximo. El nivel de dificultad se define como muestra la siguiente tabla:

<i>Nivel de Dificultad</i>	<i>Orden</i>
Fácil	0 a 35
Medio	36 a 50
Difícil	51 a 63
Muy Difícil	64

El algoritmo es el siguiente:

1. El usuario determina el nivel de dificultad.
2. Se selecciona aleatoriamente un elemento de la base de datos.
3. Una rotación aleatoria y una permutación aleatoria se aplican al elemento básico.
4. Si el nivel máximo de dificultad es escogido, imprime el elemento básico transformado; de otro modo se aplica la función “hint” a dicho elemento las veces que se necesite hasta que el Sudoku inicial alcance el orden deseado de acuerdo al nivel de dificultad requerido, y finalmente, imprime el Sudoku inicial bien definido resultante.

### 3.5. Ejemplos

Para cada nivel de dificultad se muestra un Sudoku inicial bien definido generado por el algoritmo, con su correspondiente solución.

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

(a) Nivel fácil

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

(b) Solución

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

(a) Nivel medio

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

(b) Solución

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

(a) Nivel difícil

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

(b) Solución



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

(a) Nivel muy difícil

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

(b) Solución

#### 4. CONCLUSIONES

Se realizó un análisis de la historia y la teoría matemática detrás del Sudoku y los algoritmos existentes para generar Sudokus con solución única. Se observó que estos algoritmos son muy complicados y computacionalmente, poco eficientes.

Para simplificar el proceso de generación de Sudokus con solución única, se estudió una subcolección de Sudokus iniciales de orden  $m = 64$ , que fueron llamados elementos básicos. Dichos elementos se conformaron en una base de datos de Sudokus iniciales con solución única. Este puede ser un trabajo laborioso, que puede ser considerado como una debilidad de la forma en la que el algoritmo funciona; sin embargo, de unos cuantos representantes de clases de Sudokus definidas por la relación de equivalencia de la sección 2.1, podemos generar una enorme cantidad de Sudokus distintos, pero matemáticamente equivalentes. Más aún, se catalogó el nivel de dificultad de un Sudoku de acuerdo al orden de éste, siendo que mientras mayor orden se tiene una mayor dificultad, como se observó en la sección 3.

Claramente, este proceso puede ser generalizado a Sudokus de dimensiones mayores; lo único necesario es un conjunto inicial de Sudokus con solución única.

Con este algoritmo sencillo se garantiza la unicidad de la solución, que es una condición necesaria para la creación de Sudokus válidos.

#### 5. REFERENCIAS

- [1] *The Internet ENCYCLOPEDIA OF SCIENCE: GAMES & PUZZLES.*
- [2] <http://www.freesudokupuzzle.org>
- [3] Frank, R., *Mathematics in Sudoku*, (otoño de 2005).
- [4] <http://www.cross-plus-a.com/sudoku.htm>
- [5] NIKOLI Co., Ltd.

- [6] *Springer Online Reference Works.*
- [7] Gago-Vargas, J., Hartillo-Hermoso, I., Martín-Morales, J., y Ucha-Enriquez, J. M., *Sudokus and Gröbner Bases: Not Only a Divertimento.*
- [8] **Wikipedia**
- [9] Bartlett, Andrew C., y Langville, Amy N., *An Integer Programming Model for the Sudoku Problem.*
- [10] Lee, Wei-Meng, *Programming Sudoku*, Apress, 2006.
- [11] Hayet, J.B., *Complejidad de los problemas.*
- [12] Mancini, Simona, *Sudoku Game Theory, Models and Algorithms.*
- [13] *Minimum Sudoku.*
- [14] Fasino, Dario, A “light” *Sudoku Generator and Helper.*

•  
•  
|  
|  
|  
|  
|  
|  
|  
•  
•