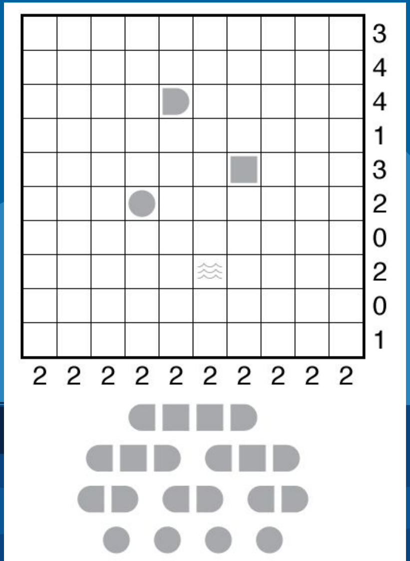
BATALLA NAVAL (PUZZLE) ES NP-COMPLETO

Equipo:

- Bonilla Reyes Dafne
- García Ponce José Camilo
- Juárez Ubaldo Juan Aurelio
- Moreno Castro Fernanda



BATTLESHIPS AS DECISION PROBLEM

Merlijn Sevenster[1]

Amsterdam, The Netherlands

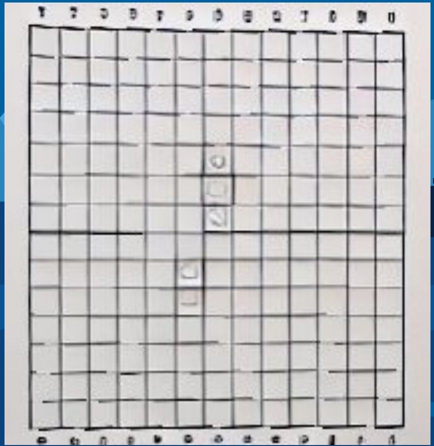
FACULTAD DE CIENCIAS, UNAM.

01

HISTORIA DEL JUEGO Y DEFINICIÓN CÓMO PROBLEMA DE DECISIÓN

HISTORIA

Apareció en revistas de rompecabezas y se volvió popular como juego lógico. Su análisis matemático lo ha relacionado con problemas difíciles en teoría de la computación, como el “Bin Packing".

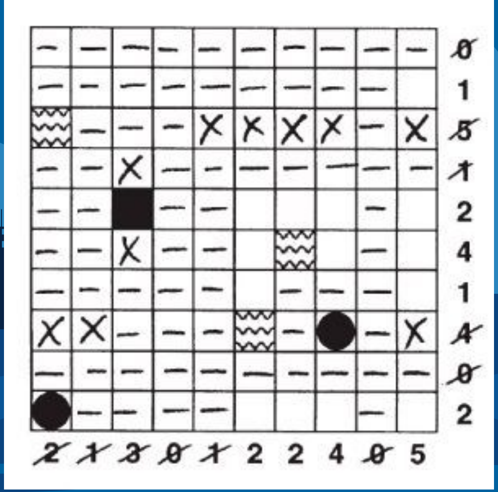


DEFINICIÓN

Un rompecabezas de Battleships de tamaño m×n (m filas y n columnas) se define como una tupla (I,C,R,F)

Condiciones de Solución (C1 a C4):

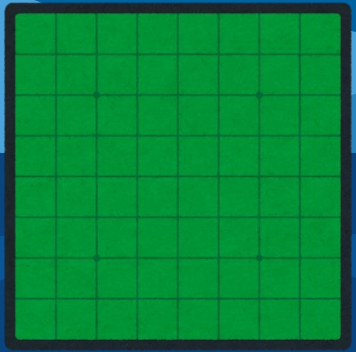
- C1: Todos los barcos de la flota deben estar en la cuadrícula.
- C2: Se deben respetar las indicaciones de la cuadrícula inicial.
- C3: Ningún par de barcos puede ocupar casillas adyacentes (ni ortogonal ni diagonalmente).
- C4: El número de segmentos de barco en cada columna (o fila) debe coincidir con el valor especificado para esa columna (o fila).



DEFINICIÓN

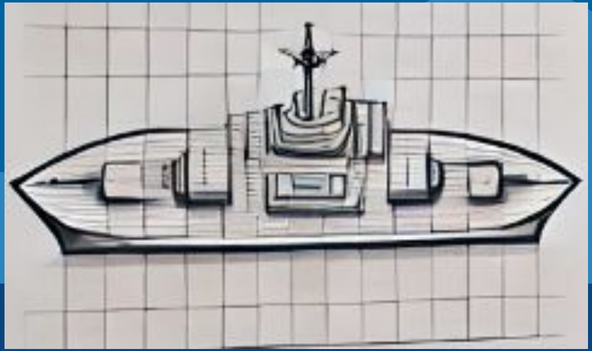
Función I: Representa el estado inicial de la cuadrícula.

- $I: \{1, \ldots, m\} \times \{1, \ldots, n\} \rightarrow \{ship, water, ?\}$
- Significado:
    - $I(i, j) = ship$: la celda (i, j) contiene un segmento de barco.
    - $I(i, j) = ?$: el contenido de la celda (i, j) es desconocido y debe decidirlo el jugador.
    - $I(i, j) = water$: en la celda sólo hay agua.



DEFINICIÓN

- **Funciones de Conteo:**
  - C: Número de segmentos de barco en cada columna.
  - R: Número de segmentos de barco en cada fila.
  - F: Flota, que indica la cantidad de barcos de cada longitud.



- **Condiciones de Solución:**
  - Existe una función J que satisface:
    - Coincidencia con I: $J(i, j) = I(i, j)$ si $I(i, j) \neq ?$.
    - Restricciones: J cumple con los conteos de C, R y F

Si J cumple estas condiciones, es una solución válida.

BATTLESHIP CÓMO PROBLEMA DE DECISIÓN

Ejemplar: una tupla I,C,R,F que representa el juego

Pregunta: ¿Existe una función J tal que cumple con las condiciones de C1 a C4 con respecto a C,R,F, ?

02

DEFINICIÓN DE REDUCCIÓN PARSIMONIOUS

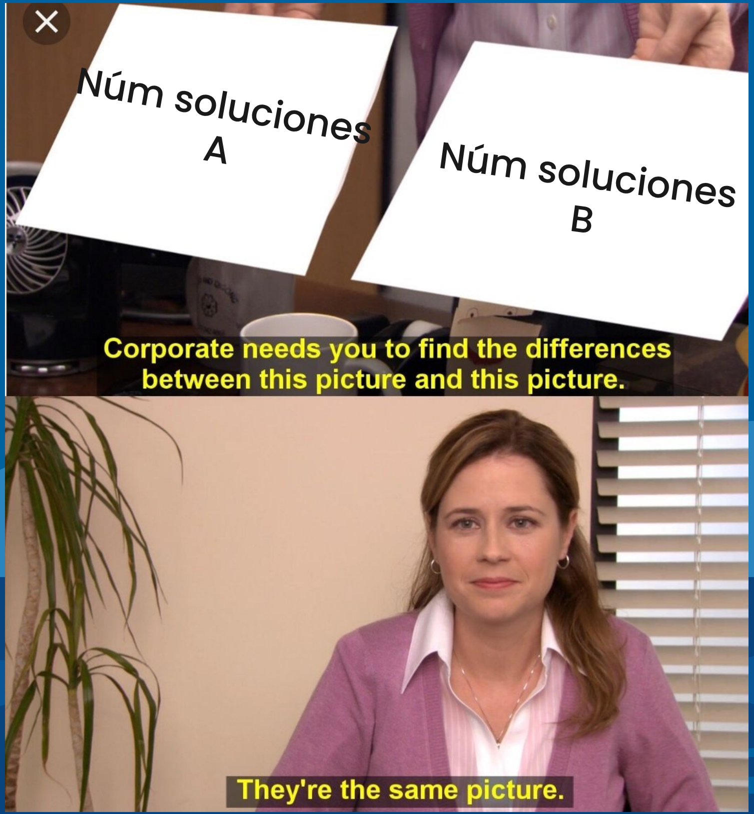
DEFINICIÓN

Definición 6.17 (reducciones parsimonious): Sea $R, R' \in \mathrm{PC}$ y sea g una reducción de Karp de $S_R = \{x : R(x) \neq \emptyset\}$ a $S_{R'} = \{x : R'(x) \neq \emptyset\}$, donde $R(x) = \{y : (x, y) \in R\}$ y $R'(x) = \{y : (x, y) \in R'\}$. Decimos que g es una reducción parsimoniosa (con respecto a R y R') si para todo x se cumple que:

$$|R(x)| = |R'(g(x))|.$$

En este caso, decimos que g es una reducción parsimonious de R a R'.

DEFINICIÓN

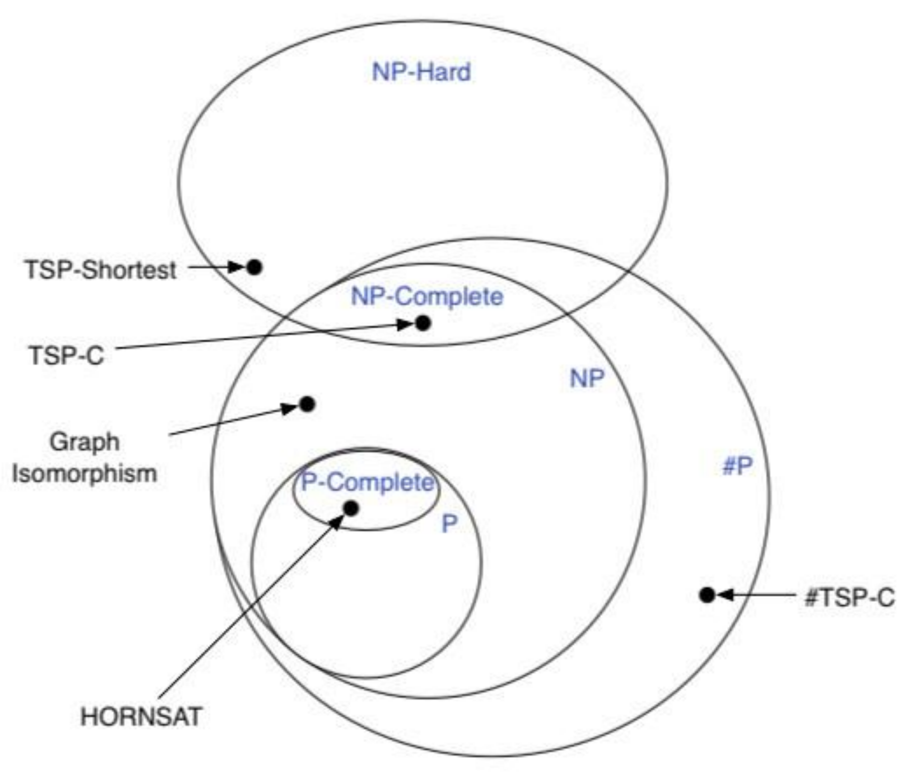

Núm soluciones A
Núm soluciones B
Corporate needs you to find the differences between this picture and this picture.
They're the same picture.

CARACTERÍSTICAS CLAVE

- Función biyectiva entre el conjunto de soluciones
- Preserva la existencia y el número exacto de soluciones
- Importante en el estudio de problemas #P-completos (#SAT)

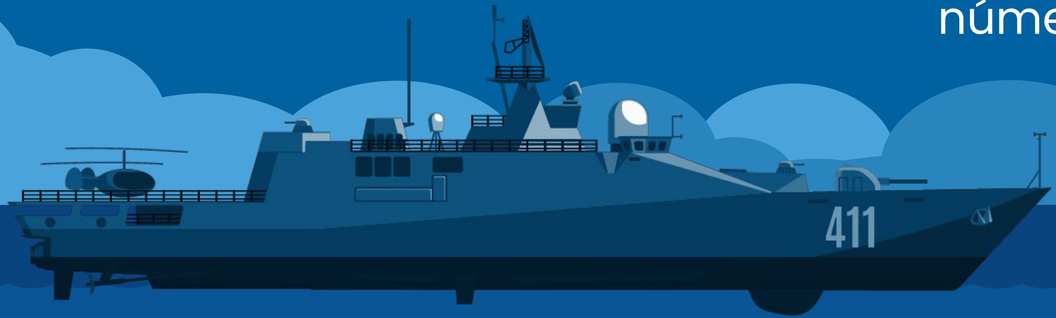
Problemas #P-completos

- Cuentan el número de soluciones a una instancia de NP
- Sus soluciones pueden ser verificadas en tiempo polinomial
- Un problema en #P es tan difícil como cualquier otro problema en #P



EJEMPLOS

- #SAT es #P-completo
- #3SAT -> Reescribimos en 3-CNF y cualquier asignación válida es válida en la forma 3-CNF, preservando el número de soluciones.



03

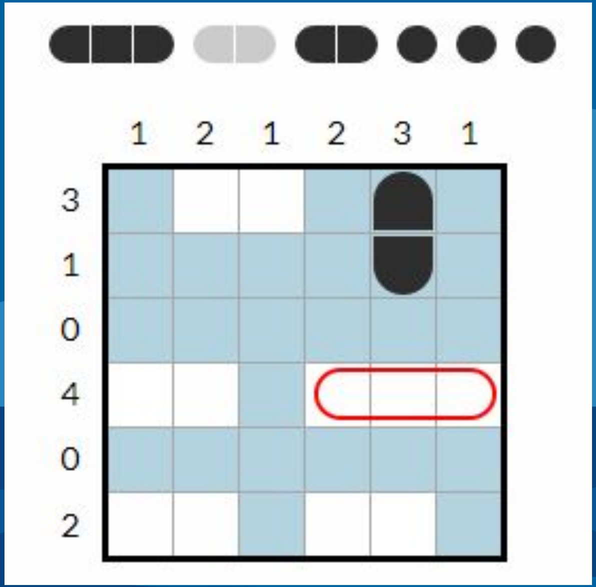
REDDUCCIÓN PARSIMONIOUS DE 3-SAT A BATALLA NAVAL

REDUCCIÓN

3-SAT → 3,{3-4}-SAT

Craig Tovey

3,{3,4}-SAT → Batalla Naval



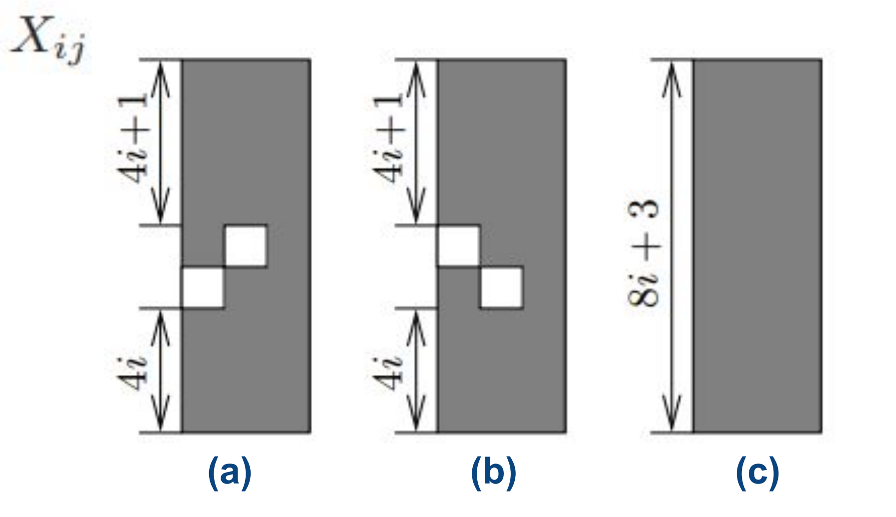
TRANSFORMACIÓN

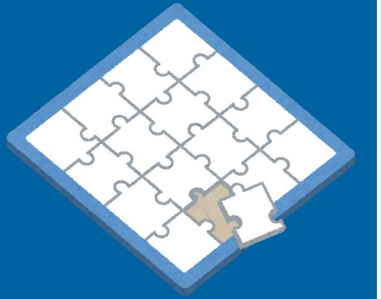
Fórmula: $\varphi \equiv C_1 \wedge \ldots \wedge C_m$

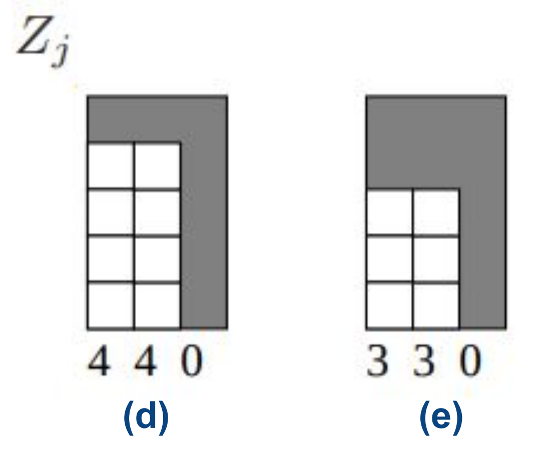
Variables: $x_1, \ldots, x_n$

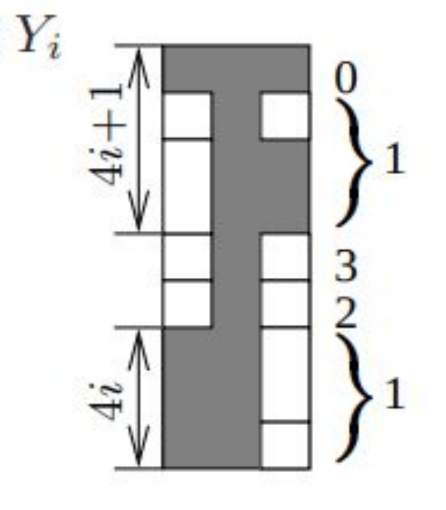
Tablero/Red:

	x_1	$\cdots$	x_n	
C_1	X_{11}	$\cdots$	X_{1n}	Y_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_m	X_{m1}	$\cdots$	X_{mn}	Y_m
	Z_1	$\cdots$	Z_n	



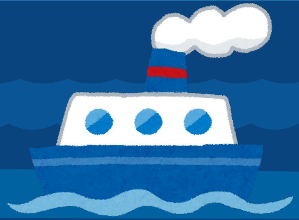






Fila	Valor
$9m+1$	0
$9m+2$	n_4
$9m+3$	n
$9m+4$	n
$9m+5$	n,

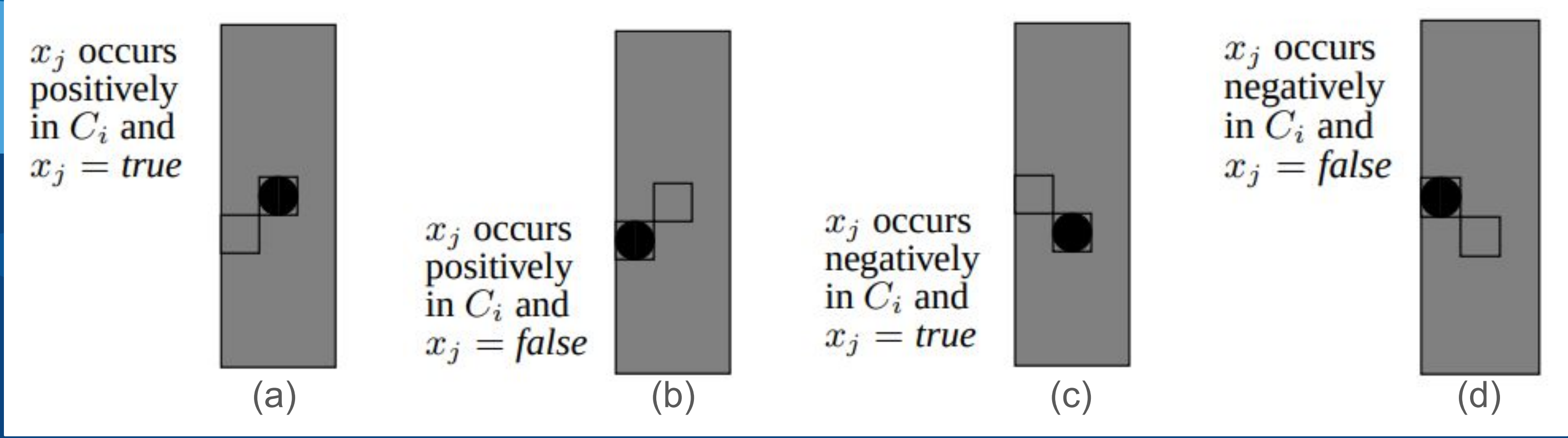
Columna	Valor
$3n+1$	$\sum_{1\leq i\leq m}(4i+1)$
$3n+2$	0
$3n+3$	$\sum_{1\leq i\leq m}(4i+1)$



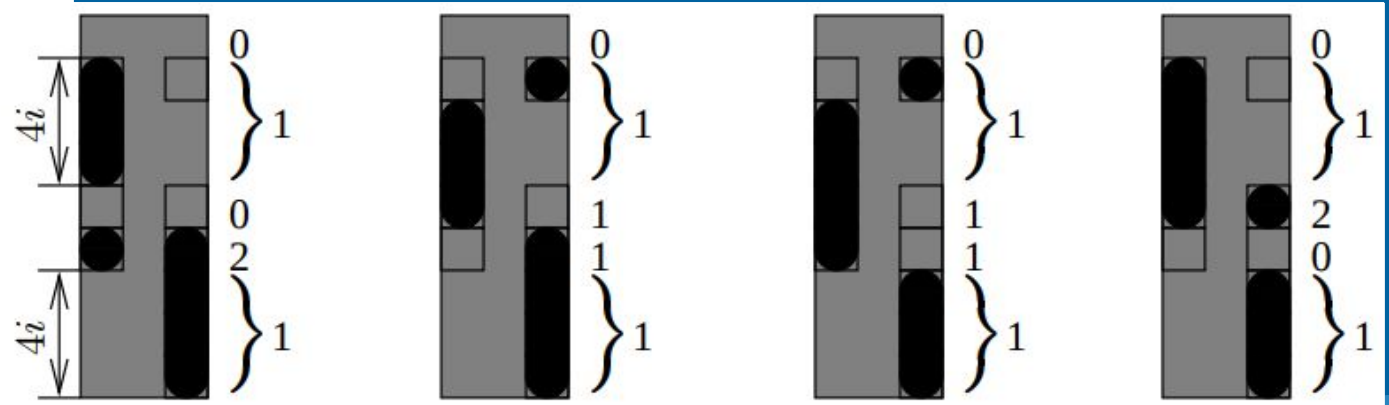
¿SOLUCIÓN?

Valores de verdad: $t : \{x_1, \ldots, x_m\} \rightarrow \{true, false\}$

X_{ij}



Y_i



(a) (b) (c) (d)

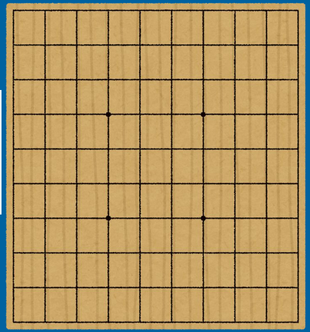
Z_j

Depende de $t : \{x_1, \ldots, x_m\} \to \{true, false\}$

Si x_j es true, entonces la nave Z_j va a la izquierda y cuando es false se pone de la derecha

Cada asignación de verdad satisfactoria genera una solución única al puzzle (y viceversa)

Casillas: $3(n + 1) * (\sum_{i=1}^{m} (8i + 3) + 5) = 12m^2 n + 12m^2 + 21mn + 21m + 15n + 15$



Naves: $nm + 3m + n$

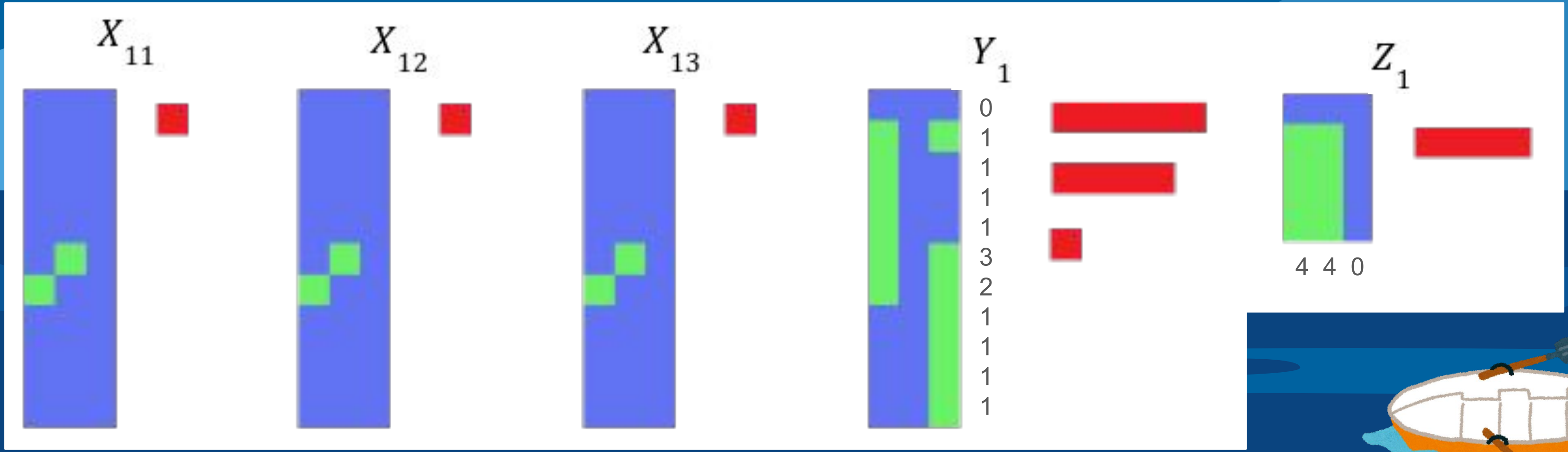
Batalla Naval es NP-Completo

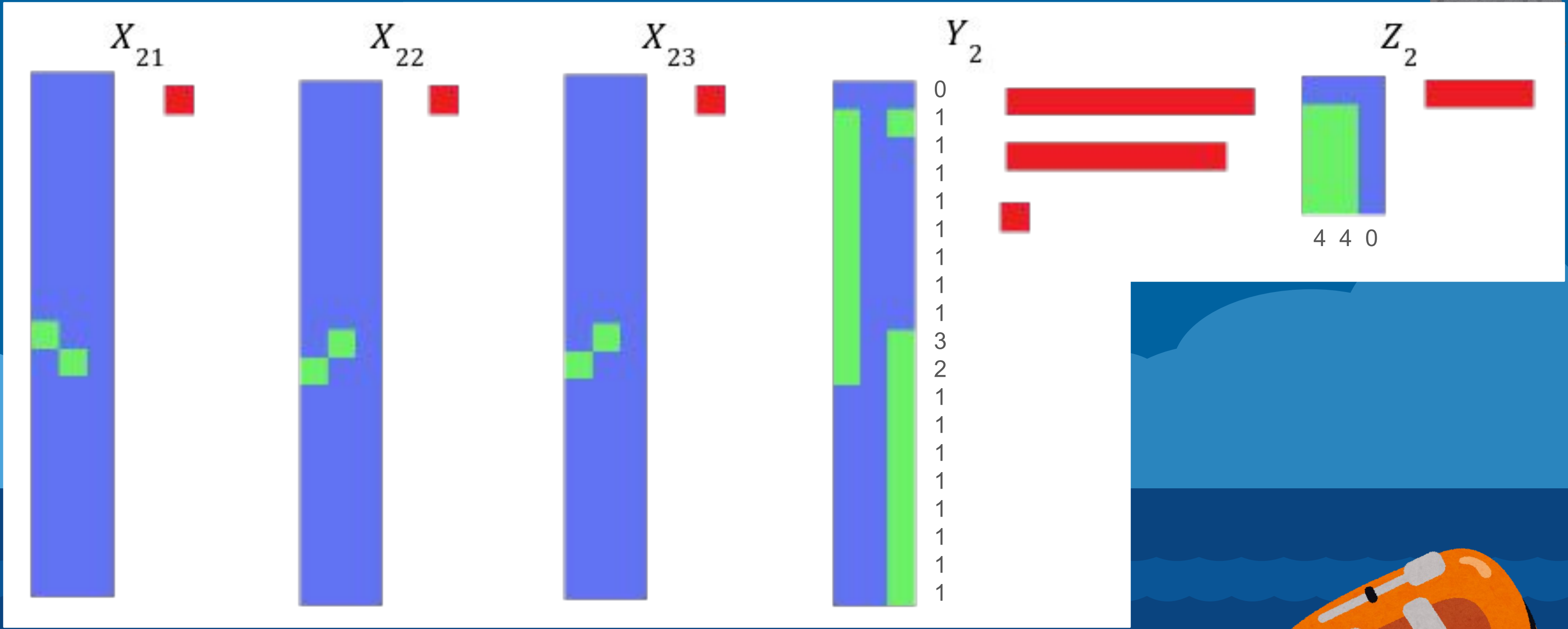
Ejemplo

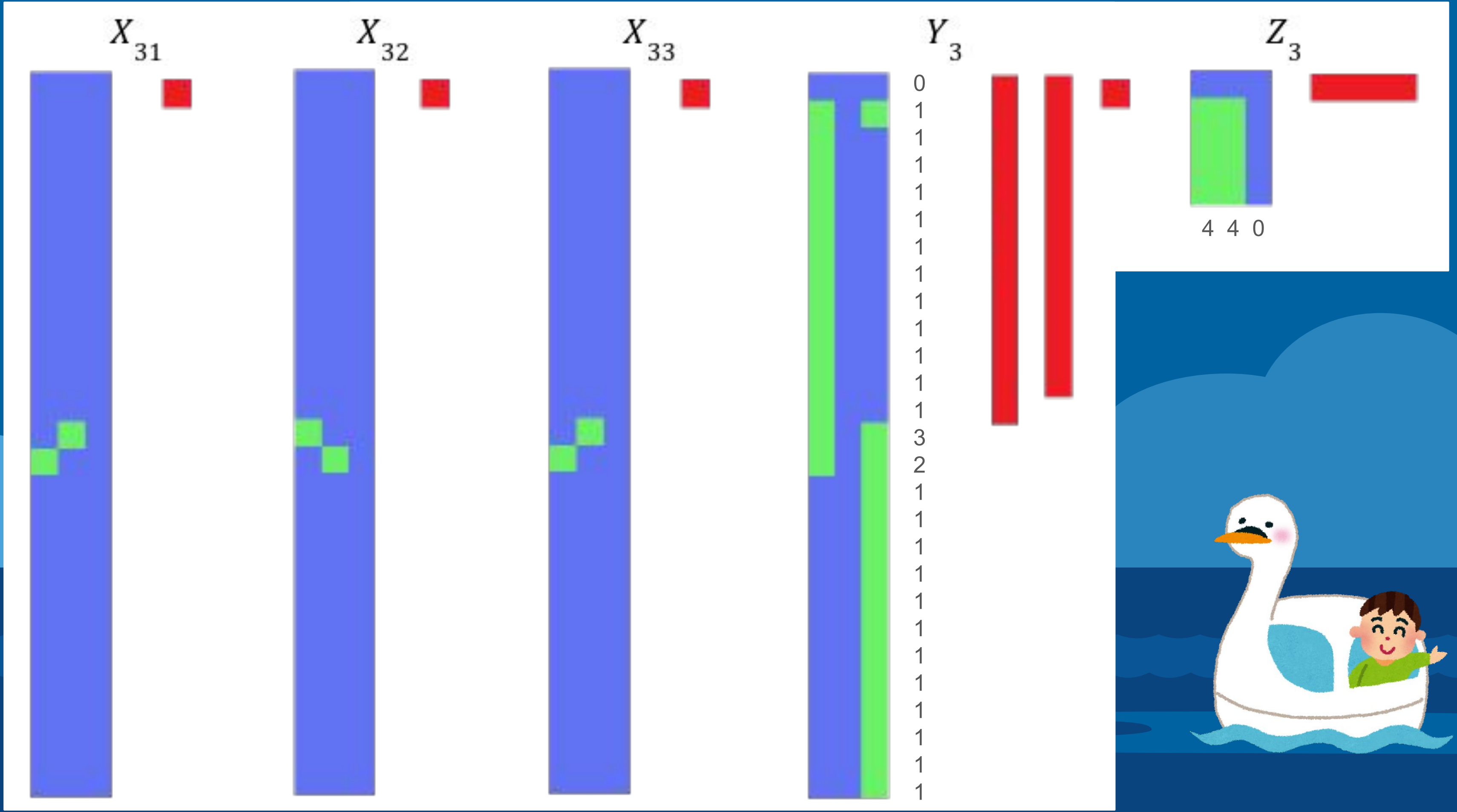
Fórmula: $\varphi = \overset{C_1}{(a \vee b \vee c)} \wedge \overset{C_2}{(\neg a \vee b \vee c)} \wedge \overset{C_3}{(a \vee \neg b \vee c)} \wedge \overset{C_4}{(a \vee b \vee \neg c)}$

Variables: $a,\ b,\ c$

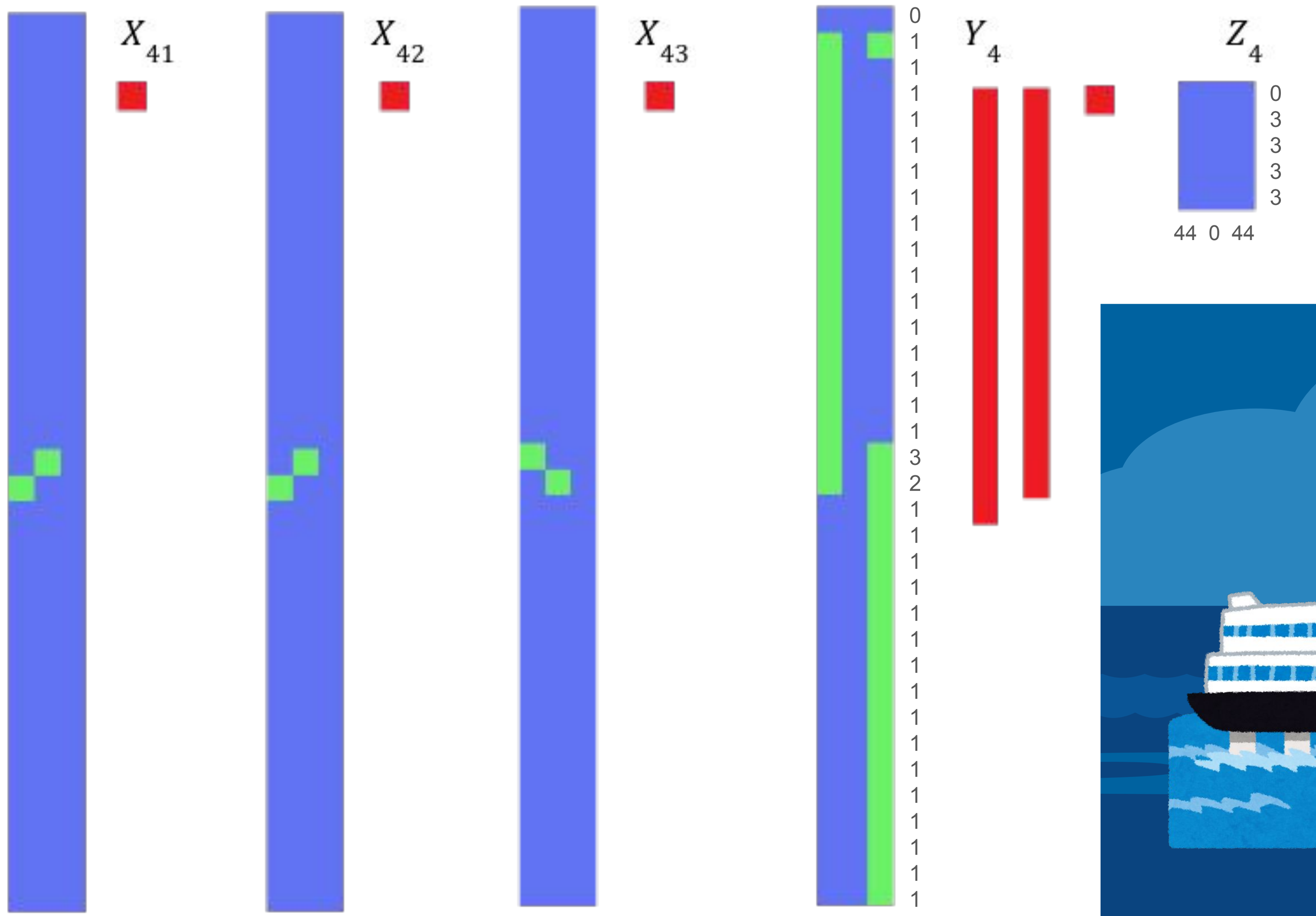
$x_1\ x_2\ x_3$



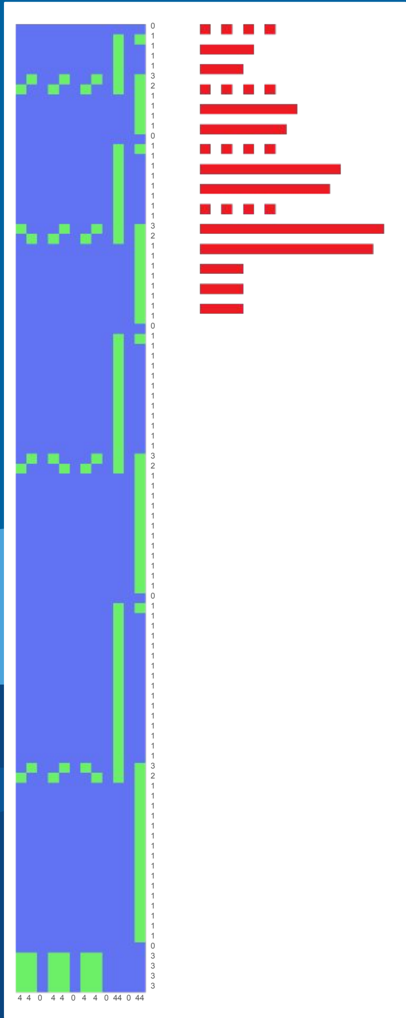

X21
X22
X23
Y2
Z2
0
1
1
1
1
1
1
1
1
3
2
1
1
1
1
1
1
1
1
4 4 0



X_{31}
X_{32}
X_{33}
Y_3
0
1
1
1
1
1
1
1
1
1
1
1
1
3
2
1
1
1
1
1
1
1
1
1
1
1
1
Z_3
4 4 0



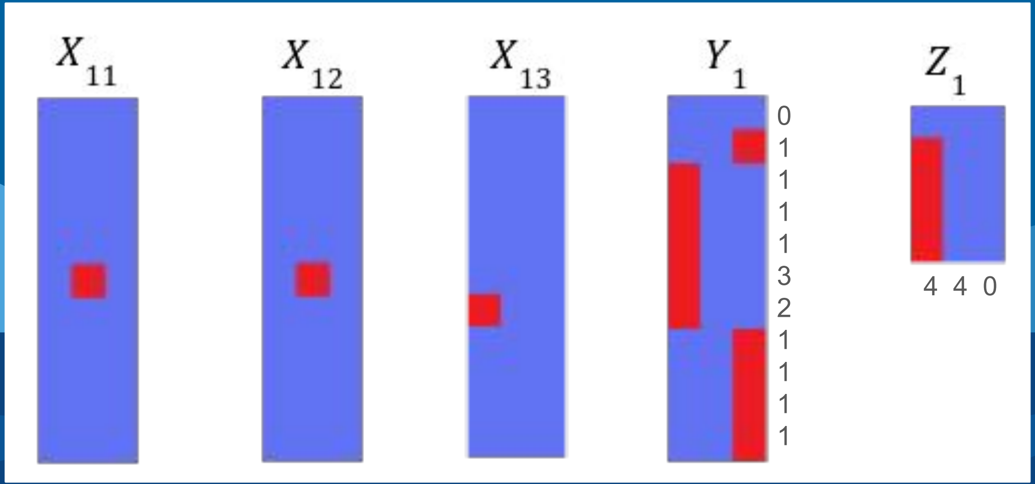
X_{41}
X_{42}
X_{43}
0
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
3
2
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
Y_4
Z_4
0
3
3
3
3
44 0 44

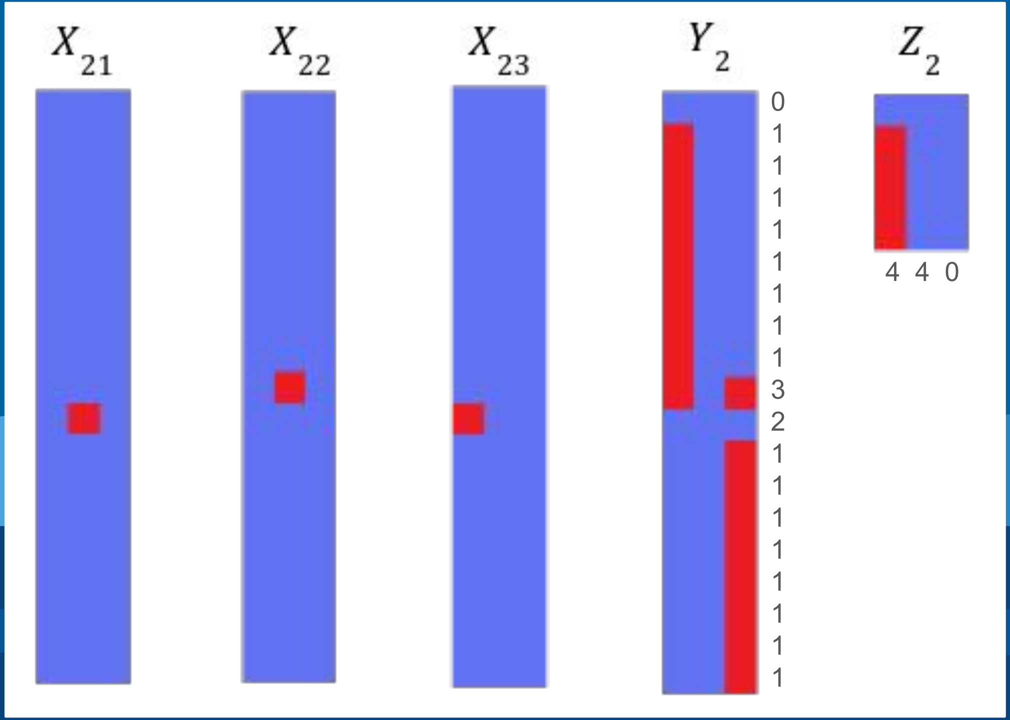


$$\varphi = (a \lor b \lor c) \land (\neg a \lor b \lor c) \land (a \lor \neg b \lor c) \land (a \lor b \lor \neg c)$$

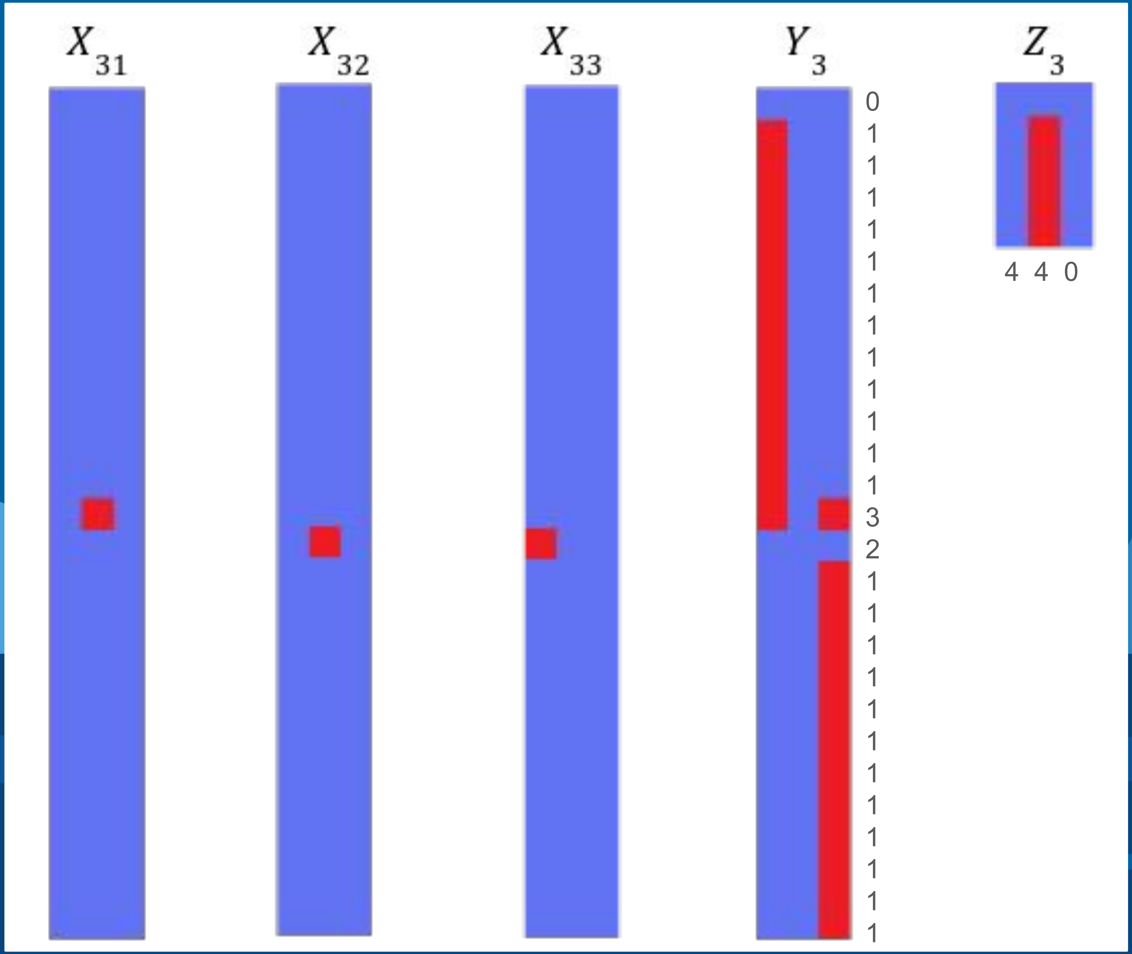
$$a = true,\ b = true,\ c = false$$

Asignación de verdad: $a = true,\ b = true,\ c = false$

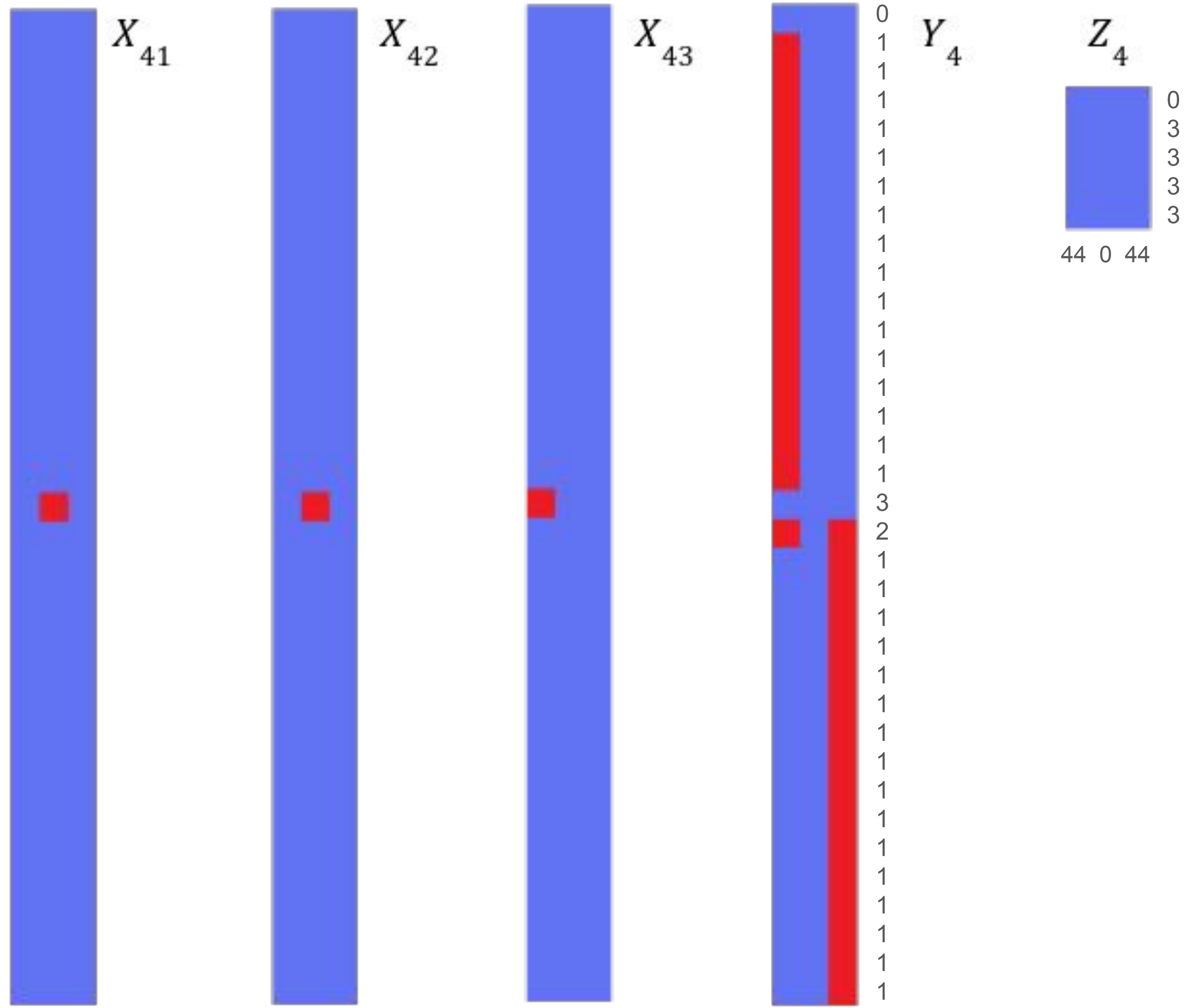



X_{21}
X_{22}
X_{23}
Y_2
Z_2
0
1
1
1
1
1
1
1
1
3
2
1
1
1
1
1
1
1
1
4 4 0

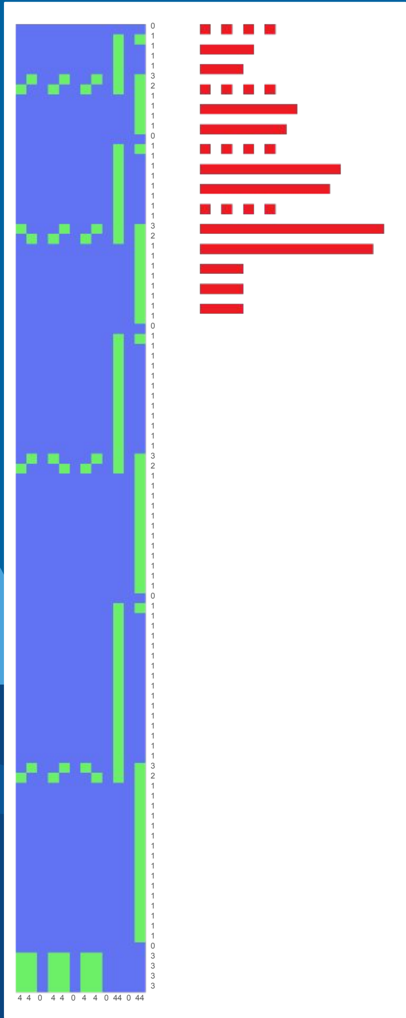

COAST GUARD

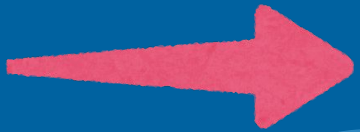


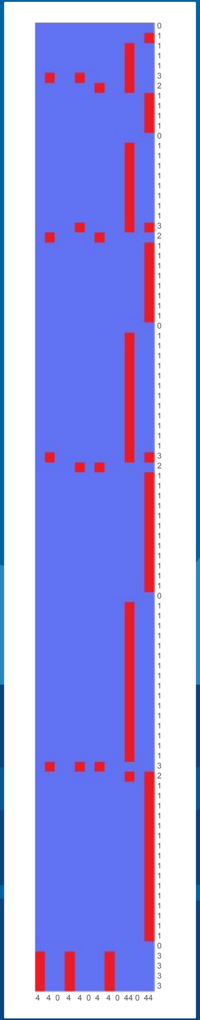
X_{31}
X_{32}
X_{33}
Y_3
Z_3
0 1 1 1 1 1 1 1 1 1 1 1 1 3 2 1 1 1 1 1 1 1 1 1 1 1 1
4 4 0

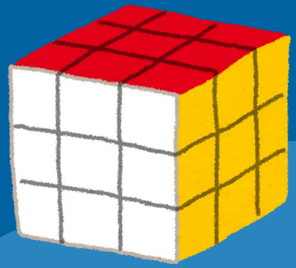


X_{41}
X_{42}
X_{43}
Y_4
Z_4
0
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
3
2
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
0
3
3
3
3
44 0 44









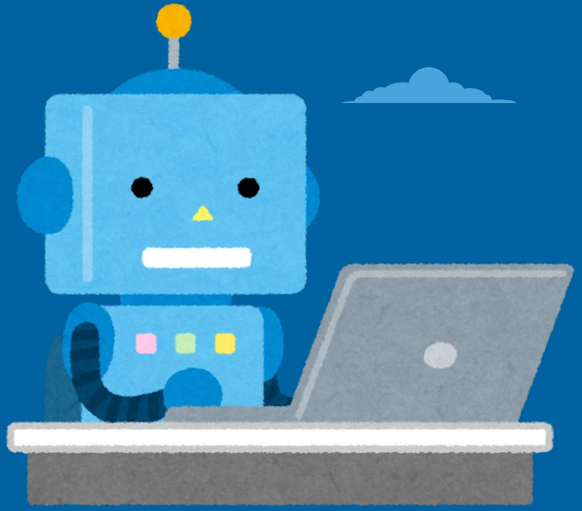

OK!

04

CONCLUSIONES E IMPLICACIONES DEL RESULTADO

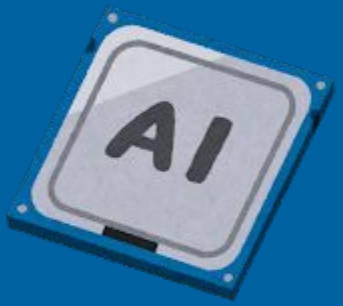
¿QUÉ CONCLUSIONES E IMPLICACIONES PODEMOS OBTENER DE ESTE RESULTADO?

1. Implicaciones prácticas
2. Implicaciones computacionales
3. Aplicaciones en otros campos



IMPLICACIONES PRÁCTICAS

Cómo vimos, este problema en general no se puede resolver de manera efectiva, por lo que el desarrollo de sistemas de inteligencia artificial puede ser clave para abordar instancias grandes o adaptadas del problema.

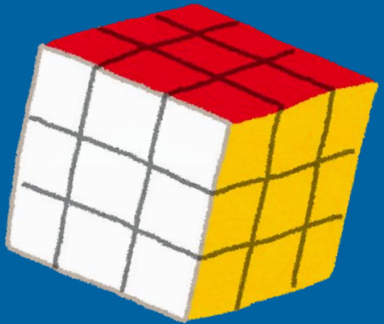


Por ejemplo, podemos hacer de algunas técnicas de IA para su resolución como:

- **Búsqueda A*:** Para explorar configuraciones prometedoras
- **Algoritmos genéticos:** Para generar poblaciones iniciales con configuraciones aleatorias y después usar selección, cruce o mutación para mejorar las soluciones
- **Redes neuronales:** Entrenar redes neuronales para predecir la disposición de barcos basándose en ejemplos resueltos

IMPLICACIONES COMPUTACIONALES

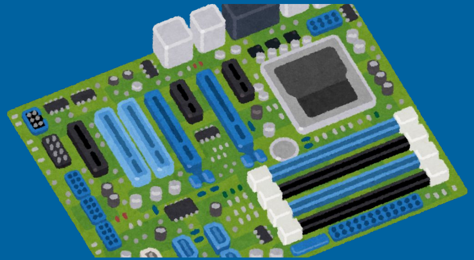
- Batalla Naval puede transformarse en problemas clásicos de la teoría de complejidad, como el problema de coloreo de grafos o el problema de empaquetado (bin packing), destacando la interconexión entre problemas NP-completos.



- Estudiar problemas NP-completos como Batalla Naval fomenta el desarrollo de métodos algorítmicos avanzados, como la programación dinámica y técnicas de reducción.

- La complejidad intrínseca del problema subraya las limitaciones actuales de los algoritmos deterministas. Sin un avance en la teoría de la complejidad, los métodos exactos serán poco prácticos.

APLICACIONES EN OTROS CAMPOS



- El problema de Batalla Naval puede modelar problemas de empaquetamiento y diseño en ingeniería (como la disposición de componentes en circuitos electrónicos o el diseño de redes).

- La dificultad del problema lo hace una herramienta excelente para enseñar técnicas de algoritmos, heurísticas y complejidad computacional.

- Las características NP-completas de Batalla Naval lo hacen ideal para diseñar juegos de lógica que sean desafiantes e intelectualmente estimulantes.


Fin.