

Sistemas de Inteligencia Artificial

Métodos de Búsqueda

Grupo 5

Juego: Skyscrapers Puzzle (Edificios)

	3	1	2	3	2	5	
2							3
2							2
3							3
1							3
2							2
3							1
	3	5	2	3	3	1	

Objetivo:

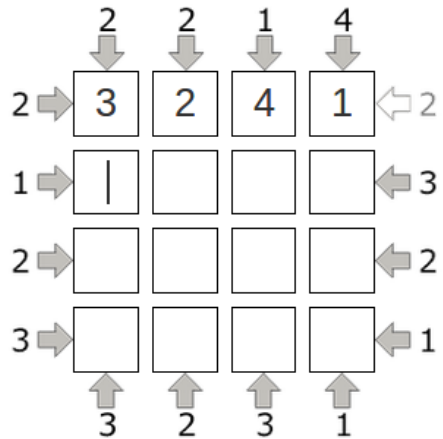
Completar la grilla con números entre 1 y N de manera que se cumpla:

- No pueden existir duplicados en fila y columna
- Se deben respetar las restricciones de visibilidad de los bordes

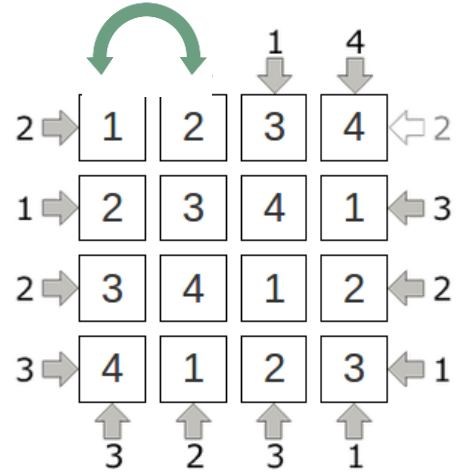
Implementación

Reglas

Fill



Swap



Heurísticas - Regla Swap

— — —

Buscan favorecer los casos donde la **cantidad de conflictos** es menor.

- Por cada **fila** que no cumpla la restricción de **visibilidad**, se puede sumar hasta 2 conflictos: 1 por la izquierda y 1 por la derecha.
- Por cada **columna** que no cumpla la restricción de **visibilidad**, se puede sumar hasta 2 conflictos: 1 por arriba y 1 por abajo.
- Por cada **columna** que tenga al menos un número **repetido** se suma 1 conflicto

Máxima cantidad de conflictos en un tablero:

$$2 * \#filas + 2 * \#columnas + \#columnas = 5N$$

Heurísticas - Regla Swap

— — —

Para no sobreestimar el costo real y encontrar el camino mínimo, se define la máxima cantidad de conflictos que se pueden resolver en 1 swap:

Intercambiando dos casilleros horizontalmente: **2** por la **visibilidad de la fila** involucrada + **2** por la **visibilidad de cada columna** involucrada + **1** por los **posibles repetidos de cada columna** involucrada.

Máx. cantidad de conflictos que se pueden resolver con 1 swap = MCCRS:

$$2 + 2 * 2 + 1 * 2 = 8$$

Heurística Admisible - Regla Swap

— — —

$$\begin{aligned} h(n) &= \lceil \# \text{conflictos del tablero} / \text{MCCRS} \rceil * \text{costo del swap} \\ &= \lceil \# \text{conflictos del tablero} / 8 \rceil \end{aligned}$$

Heurística No Admisible - Regla Swap

— — —

$$h(n) = \text{\#conflictos} * \text{costo del swap} = \text{\#conflictos}$$

Resultados

Resultados con regla FillRule

	3	0	1	
3	0	0	0	1
0	0	0	0	0
1	0	0	0	2
	0	0	2	

Tablero 3x3

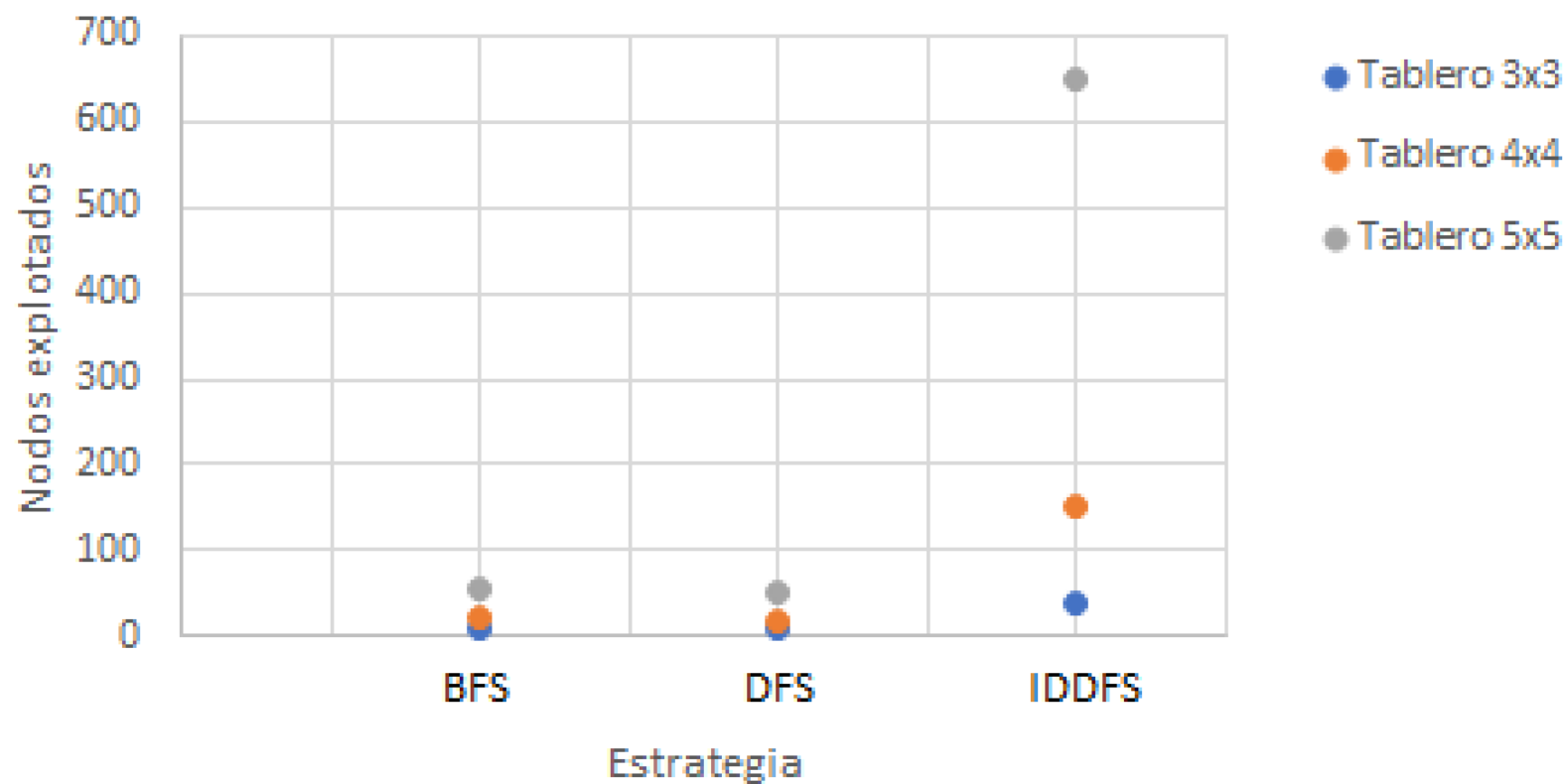
	4	0	2	1	
4	0	0	0	0	1
3	0	0	0	0	2
0	0	0	0	0	0
1	0	0	0	0	2
	1	2	0	2	

Tablero 4x4

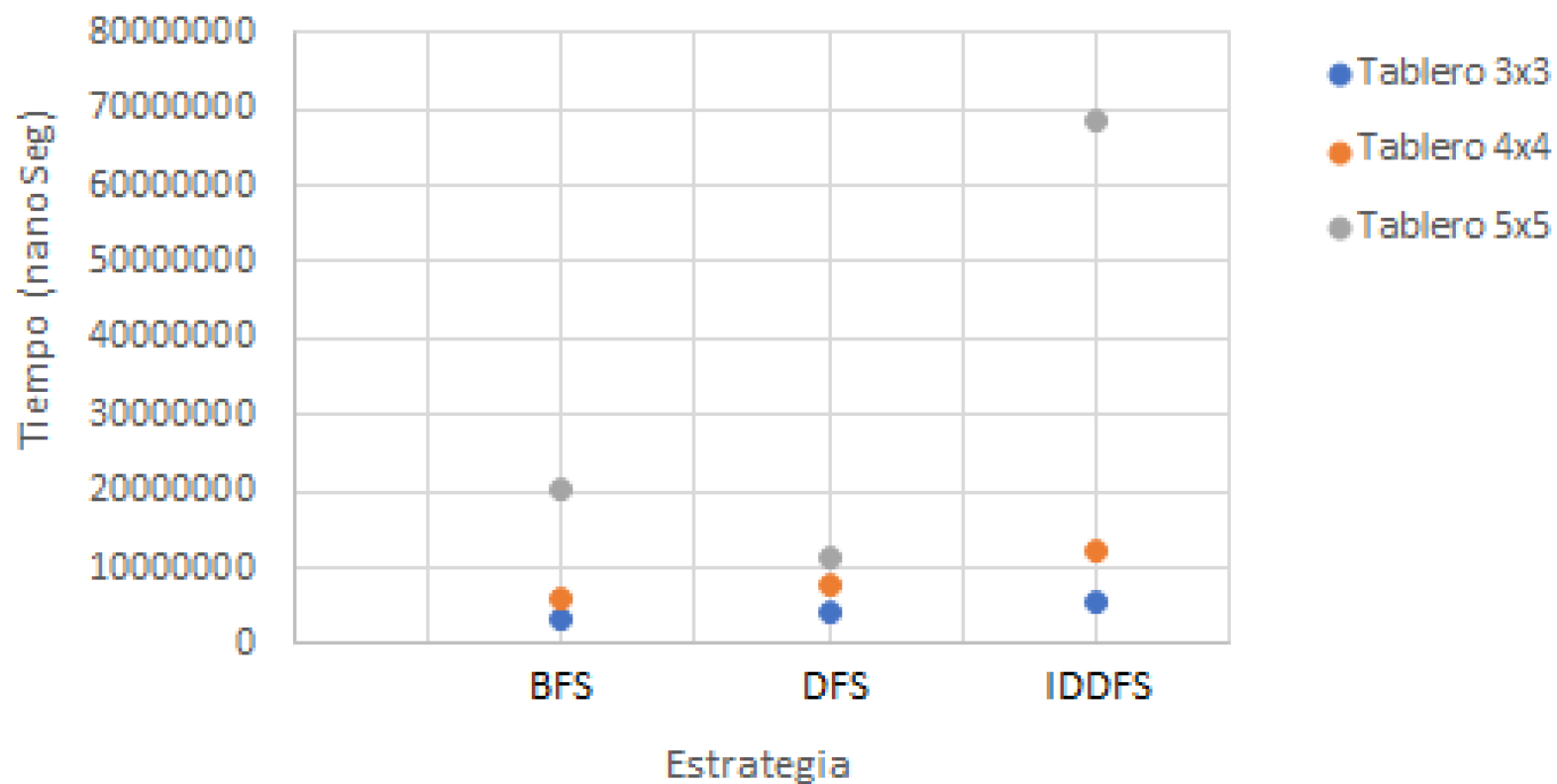
	5	4	3	2	1	
5	0	0	0	0	0	1
0	0	0	0	0	0	0
3	0	0	0	0	0	2
2	0	0	0	0	0	2
0	0	0	0	0	0	0
	1	0	0	2	2	

Tablero 5x5

Fill Rule



Fill Rule



Resultados con regla SwapRule

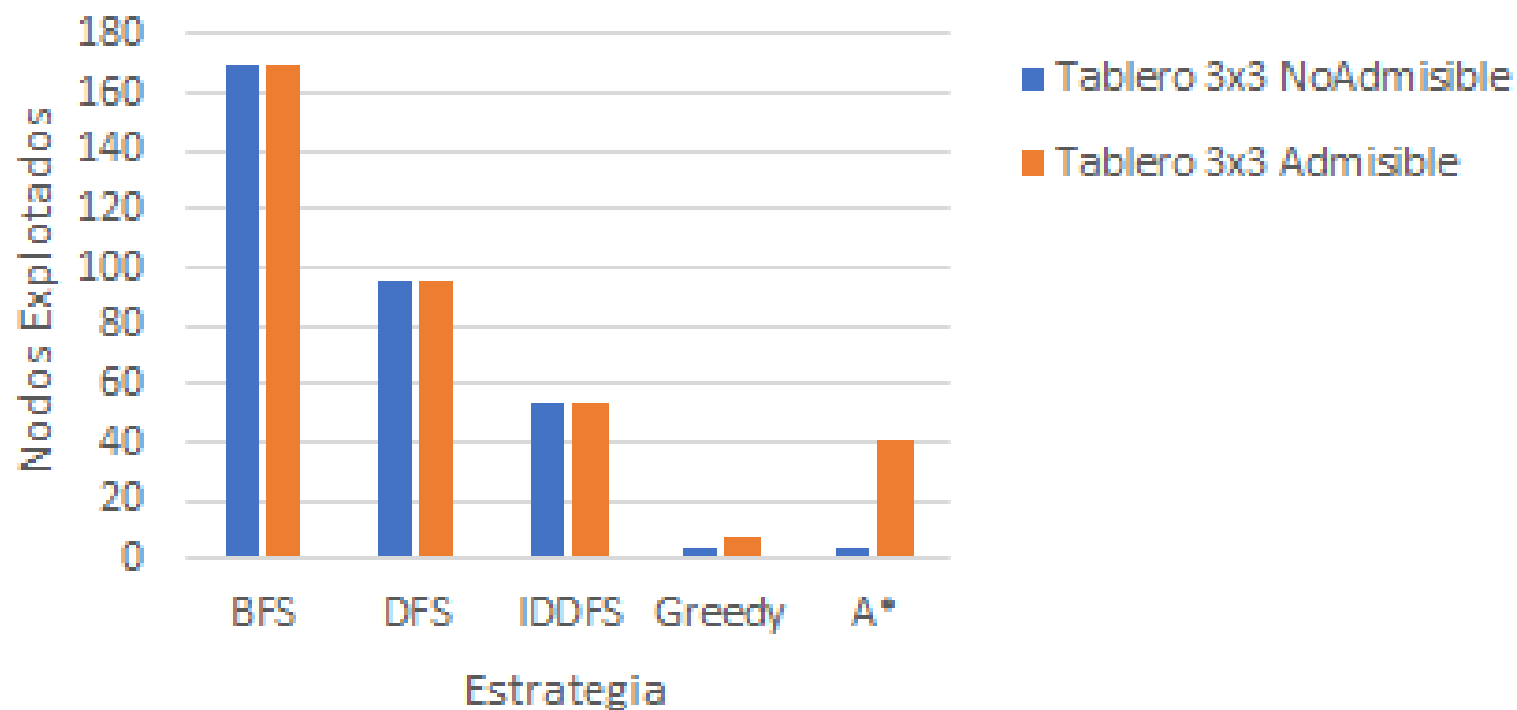
	2	2	1	
3	1	2	3	0
1	1	2	3	2
2	1	2	3	2
	0	0	3	

Tablero 3x3

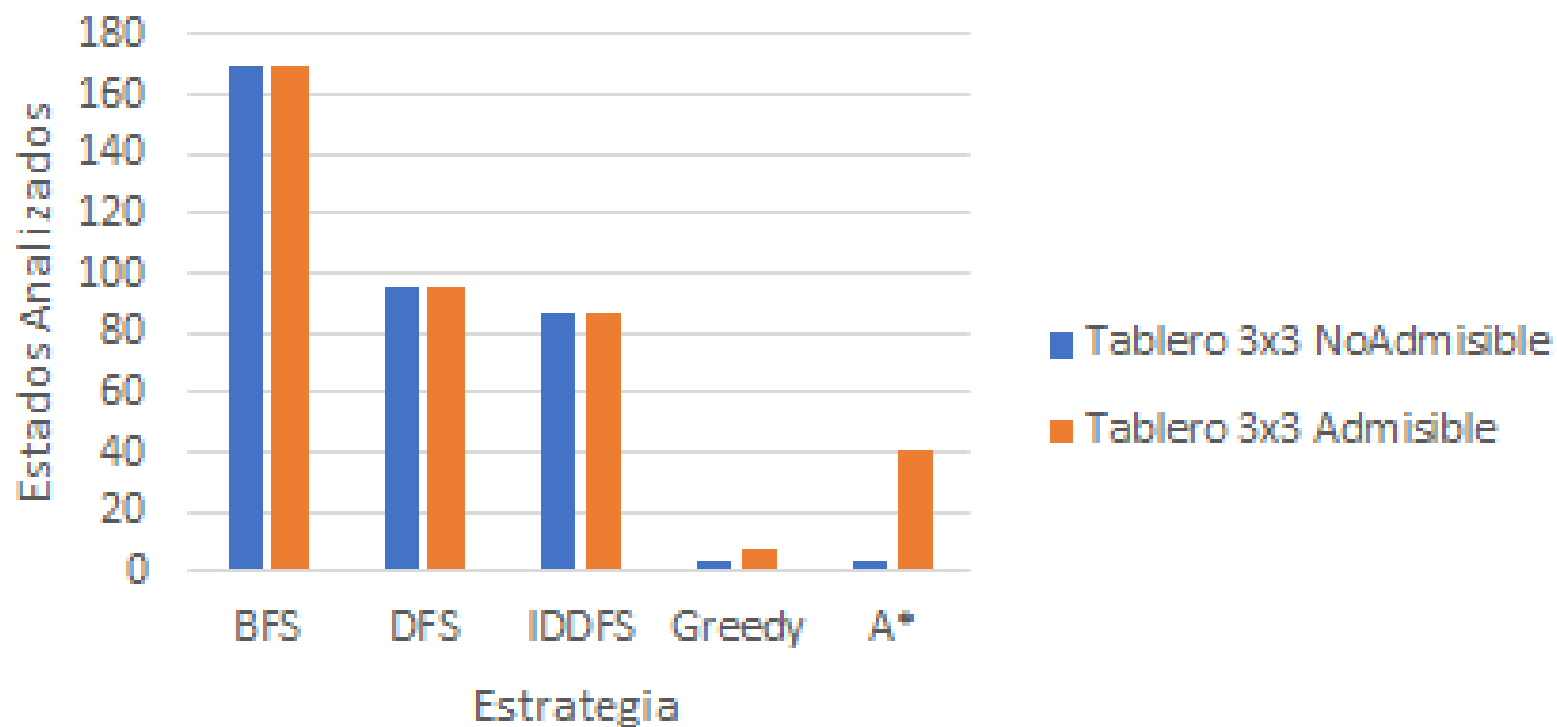
	3	2	2	1	
4	1	2	3	4	1
2	2	3	4	1	2
1	3	4	1	2	4
2	4	1	2	3	2

Tablero 4x4

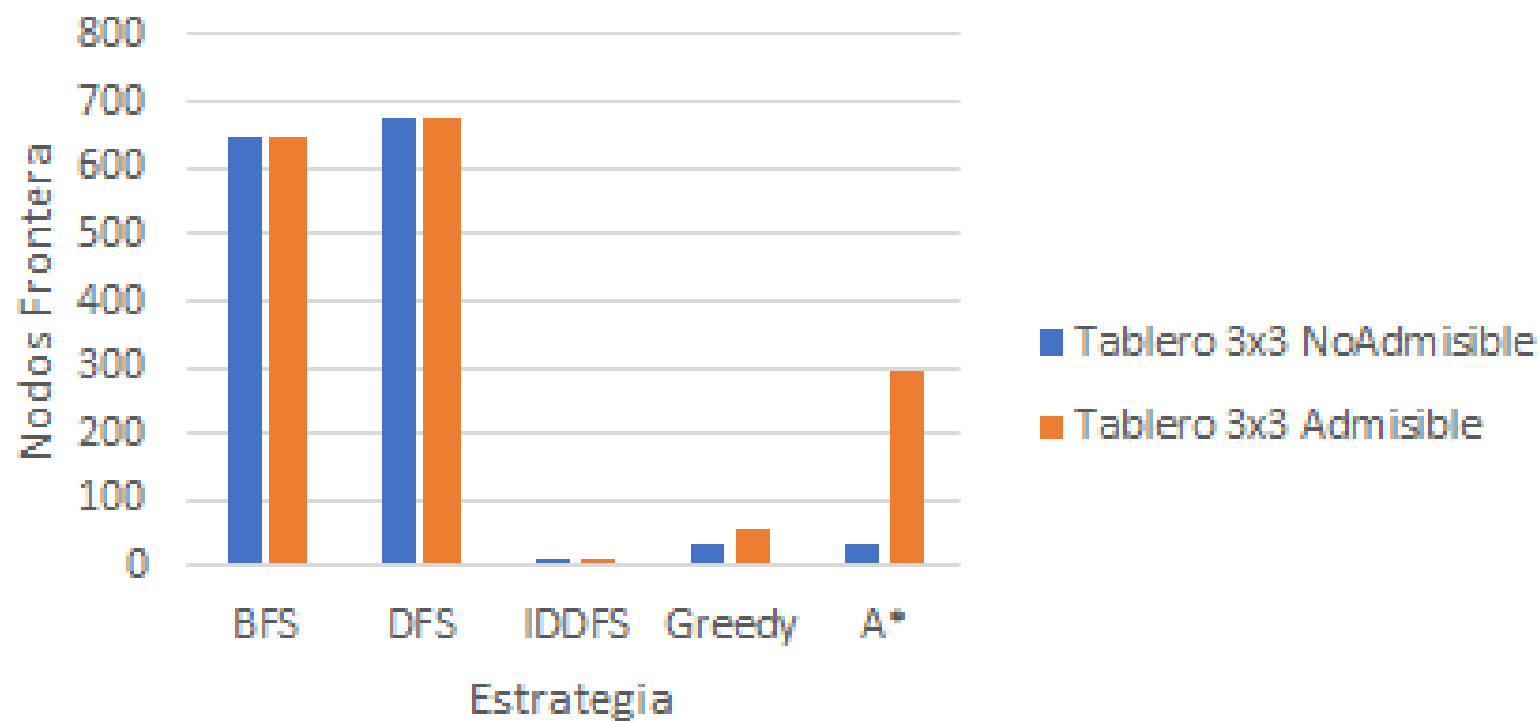
Swap Rules 3x3



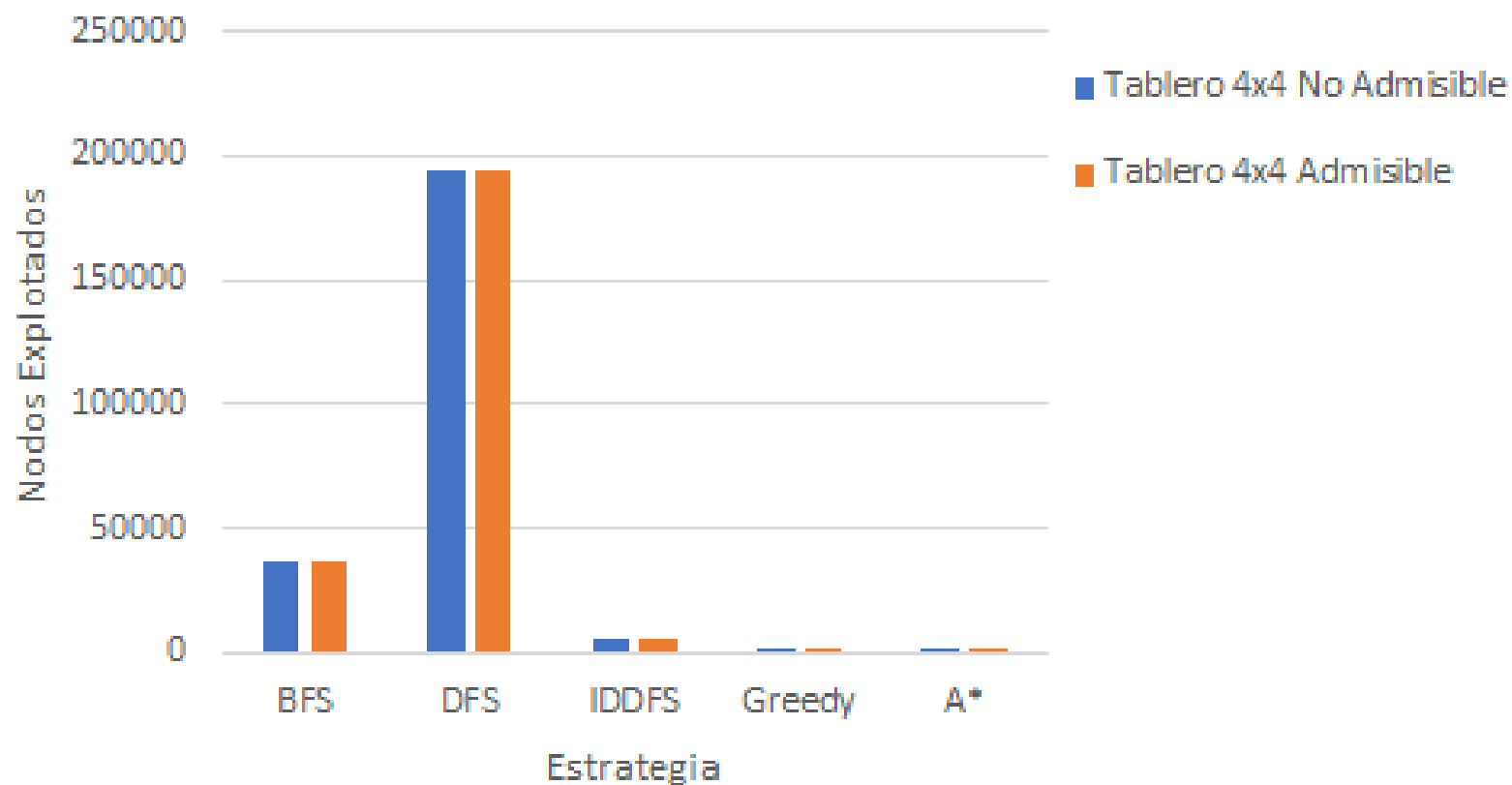
Swap Rules 3x3



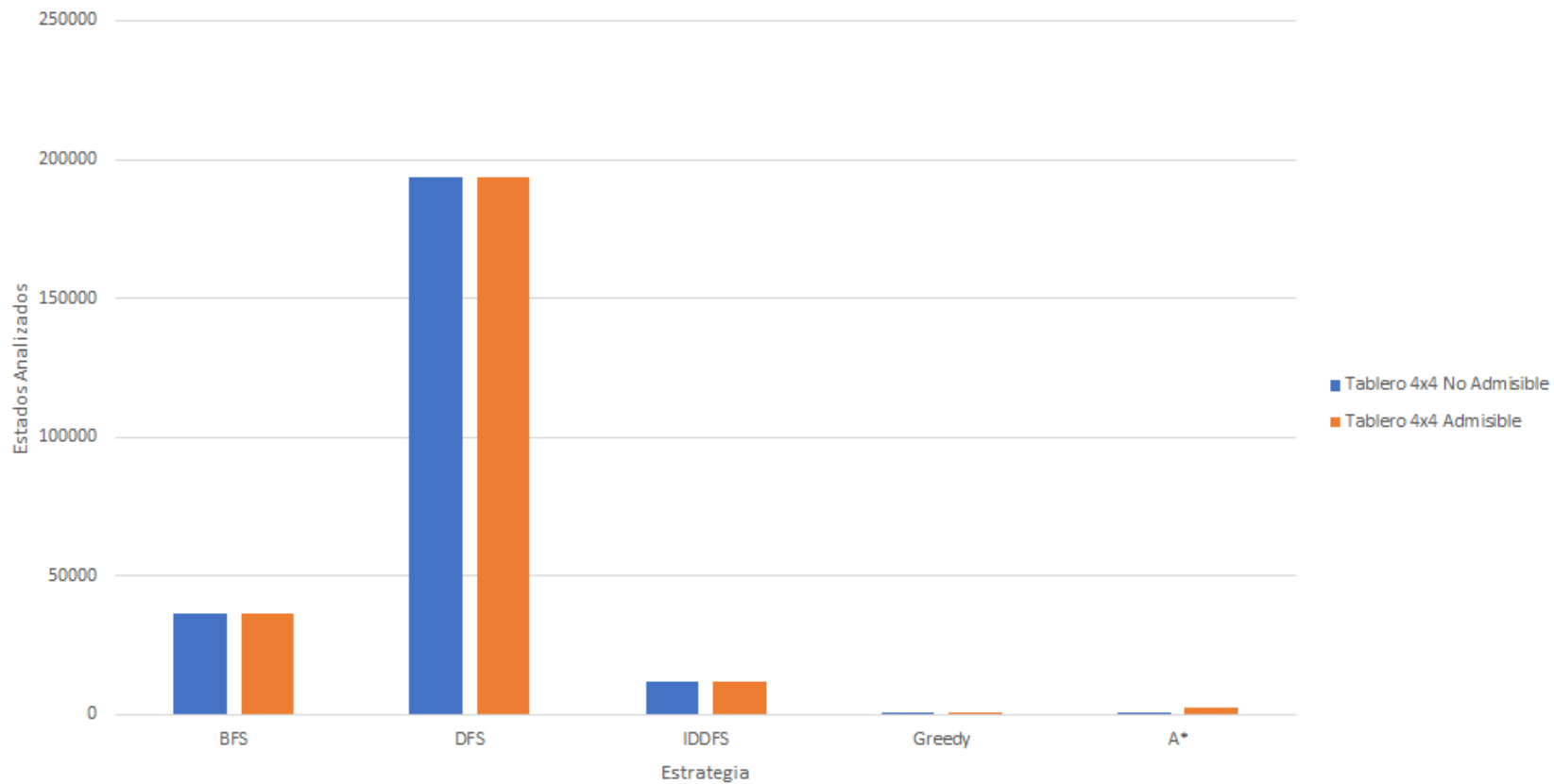
Swap Rules 3x3



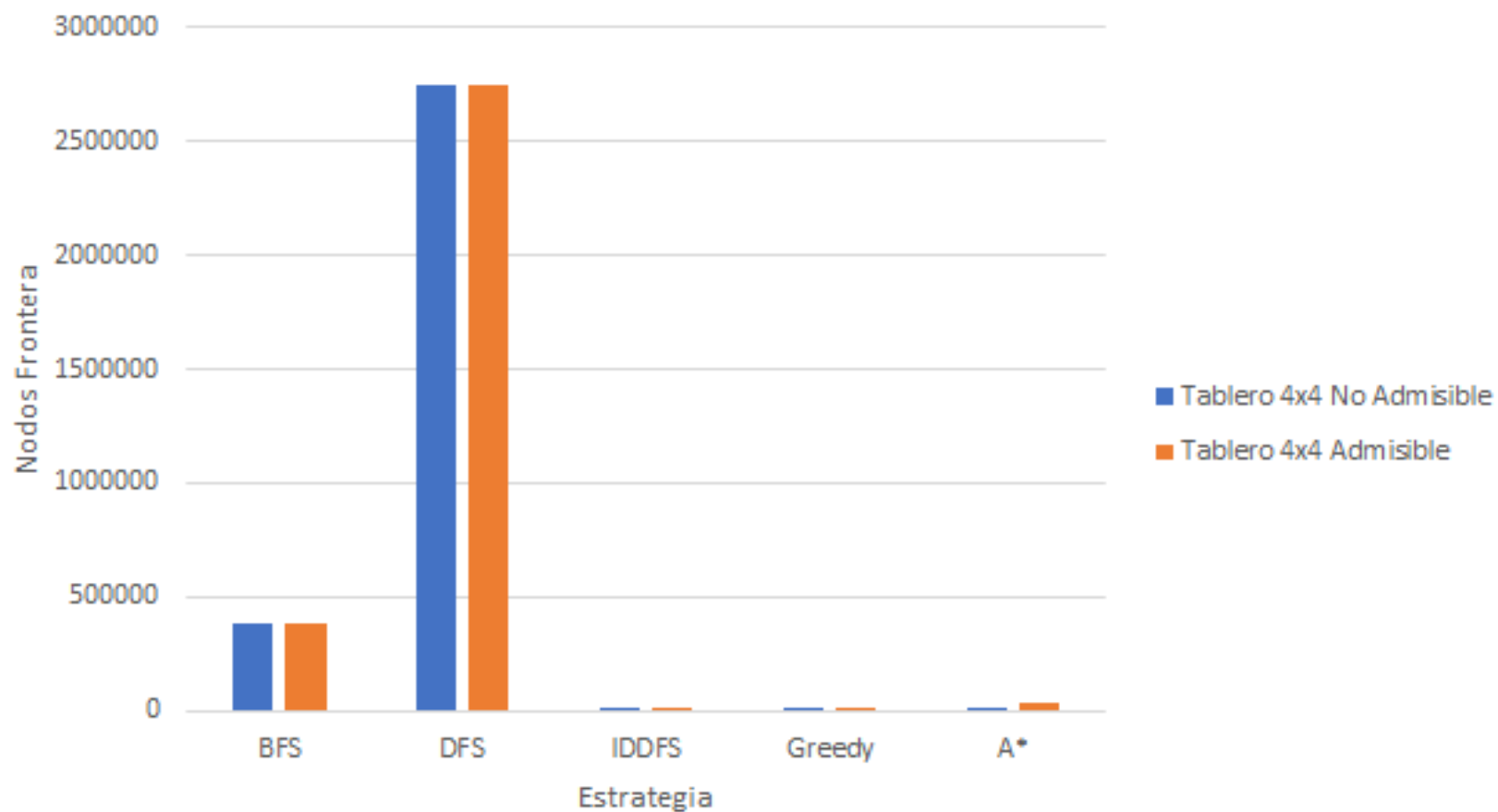
Swap Rules 4x4



Swap Rules 4x4



Swap Rules 4x4



Conclusiones

Conclusiones

— — —

- Luego de analizar los tiempos de ejecución, se puede ver claramente que **definir un buen costo** de aplicar una regla es esencial, ya que permite llegar a una solución mucho más rápidamente y con menos esfuerzo.
- Utilizar un **mapa de estados** es clave para que los algoritmos puedan terminar correctamente y no quedarse en loops infinitos. Además, se reduce el tiempo de ejecución drásticamente.

Conclusiones

— — —

- Mejor tiempo de ejecución en búsqueda no informada: **DFS** (FillRule)
- Definir bien el **hash** y el **equals** para todas las clases necesarias es imperativo para que el algoritmo de búsqueda funcione bien y pueda determinar correctamente cuáles estados son exactamente iguales y cuales no.

Mejoras posibles

¿Preguntas?

Fin