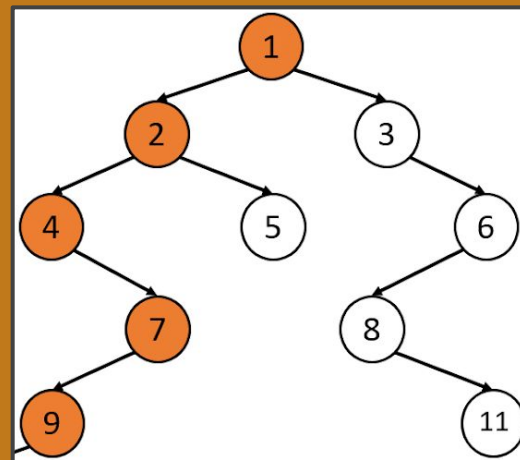


Trabajo Práctico N°1

Métodos de Búsqueda

Grupo 7:

- ❖ Valentin Ye Li
- ❖ Nicolás Birsa
- ❖ Andres Podgorny
- ❖ Oussama Slaitane
- ❖ Hugo Lichtenberger



8-Puzzle

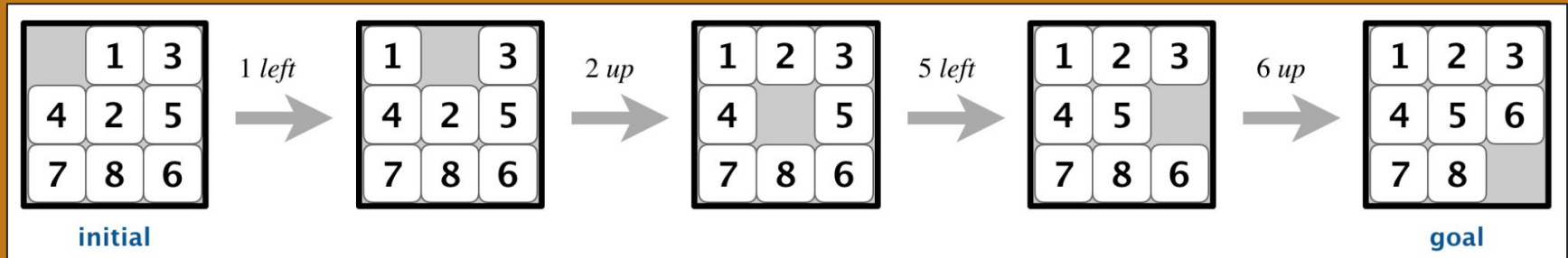
- ❖ Reglas
- ❖ Estructura de estado
- ❖ Heurística y Método a usar



Reglas del juego

Consiste en un tablero de 3x3 con 8 casillas numeradas y una casilla vacía. El objetivo es mover casillas hasta llegar a un tablero solución.

En cada paso se desplaza una casilla hacia el espacio vacío contiguo, pudiendo moverla en vertical (arriba y abajo) o en horizontal (derecha e izquierda).



Estructura de estado

Se plantea una matriz 3x3, utilizando casillas con valores en [1, 8]. La casilla vacía se representa con algún valor no presente en [1, 8], como 0 o *None*.

Dos estados son **iguales** si ambas matrices tienen los mismos valores en cada posición.

Heurística 1 - Distancia Manhattan

La suma de la **cantidad de movimientos mínimos** directos de cada bloque hasta su lugar correspondiente.

Es **admisibles** por que nunca sobreestima el costo real, pues considera movimientos “ideales”.

$$H_1 = M(1) + \dots + M(8) = 11$$

5	7	3
8	2	
1	6	4

11 pasos

1	2	3
8		4
7	6	5

Heurística 2 - Manhattan mejorado

Usa de base la Heurística anterior, pero considerando que si hay **dos bloques adyacentes con su posición final invertida**, se deben realizar como mínimo dos movimientos extra para llegar a la posición final.

Es **admisable** porque agrega un caso real a movimientos ideales.

5	1	3
7	2	
8	6	4

Bloques 7 y 8 están invertidos

$$\begin{aligned} H_1 &= 7 \\ H_2 &= H_1 + 2 = 9 \end{aligned}$$

Método y Heurística a usar

Método elegido: A*

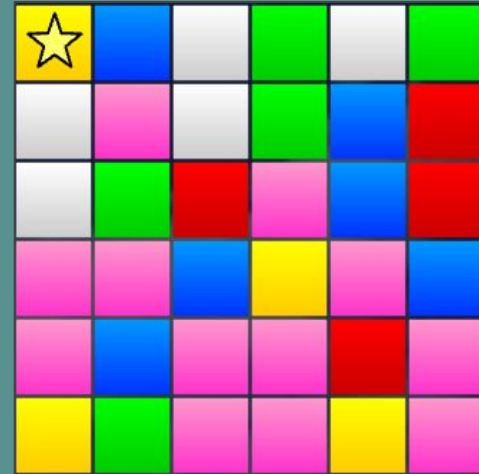
- ❖ BFS y DFS no pueden estimar en runtime el mejor camino a seguir, que significa o mayor demora al buscar el camino u obtener uno poco óptimo.
- ❖ Greedy Search solo usaría la heurística provista, por lo que, de ser poco precisa en comparación al costo real, podría llegar a una solución menos óptima.

Heurística elegida: Manhattan mejorado

- ❖ A pesar de que tanto la Distancia Manhattan como su versión mejorada estiman el mismo costo en el ejemplo provisto, en un caso genérico conviene usar la segunda al contemplar más casos y poder devolver un costo estimado mayor, sin dejar de ser admisible.

Fill-Zone

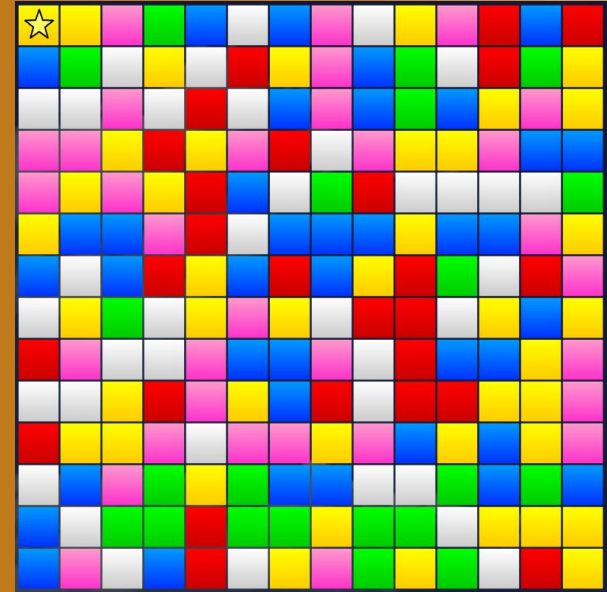
- ❖ Reglas
- ❖ Estructura de estado
- ❖ Heurísticas usadas
- ❖ Resultados



Reglas del juego

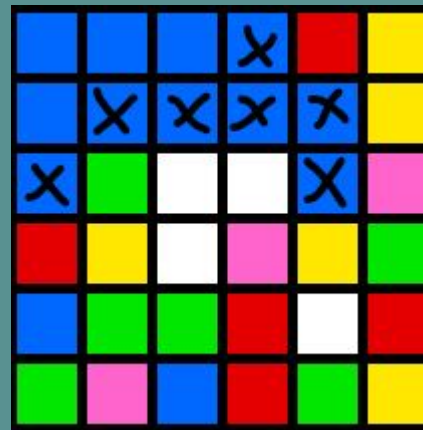
El juego tiene como objetivo colorear todas las casillas de un tablero de **NxN** con el mismo color.

Se empieza desde la esquina superior izquierda, y con cada cambio de color las casillas de ese nuevo color que estén en contacto con la “zona activa” pasan a formar parte de esta.



Estructura de estado

- Para representar la grilla se guarda una matriz de $N \times N$, pero para resolver el juego sólo se recorren las celdas “*frontera*” de la zona activa.
- Las celdas de la zona activa que no son frontera se marcan y no se vuelven a recorrer.
- A la hora de expandir la zona de bloques, se recorre la frontera en búsqueda de casillas adyacentes del nuevo color, que se agregan a la nueva frontera.



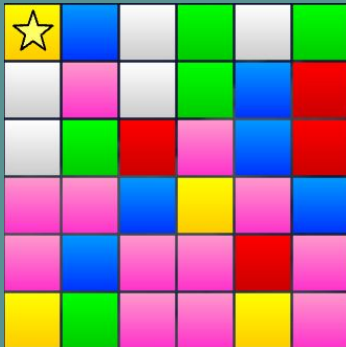
Consideraciones

1. El problema se toma de **costo uniforme** ya que todas las acciones tienen el mismo “costo”, que es cambiar a otro color la zona activa.
2. En todas las mediciones se usaron 6 colores, como es en el juego.
3. No se simuló la acción de shuffle para asegurar la consistencia de los resultados.
4. Para las pruebas se usó una cantidad de turnos máxima de 1000, que es más que suficiente para poder encontrar una solución en un tablero de 9x9.

Heurística 1 - Promedio de cantidad de casillas de cada color

Para cada color en el tablero, se cuenta cuántas casillas tienen dicho color y se divide entre el total de casillas. Así, la heurística es la suma de las fracciones obtenidas para cada color.

Es **no admisible** ya que el tablero puede ser inicializado de un mismo color, donde la heurística estimaría un paso para terminar cuando ya se ganó el juego.



$$H_1 = \#_{\text{azul}} / \#_{\text{total}} + \dots + \#_{\text{rojo}} / \#_{\text{total}} = \frac{1}{1}$$

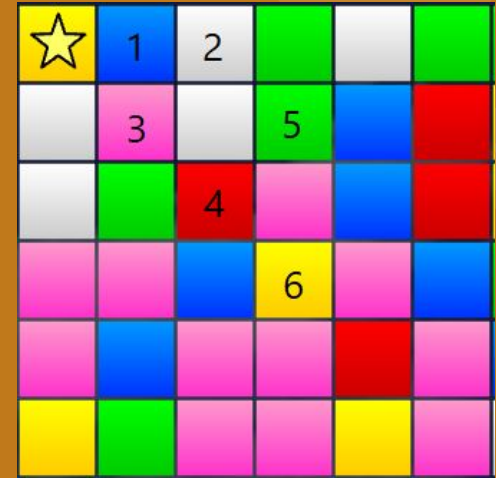
Heurística 2 - Cantidad de colores restantes

Se cuentan la cantidad de colores diferentes presentes en el tablero, sin tener en cuenta la casilla inicial.

Es **admisible** ya que para cualquier tablero se requiere como mínimo cantidad de cambio de colores igual a la cantidad de colores diferentes.

Para el tablero de ejemplo (y en general):

$$H_2 = M \text{ (siendo } M = \#_{\text{colores}} \text{)}$$



Heurística 3 - Distancia de Bronson

Estima la cantidad de cambios de color que hay que hacer para terminar el juego en función de la celda inferior derecha.

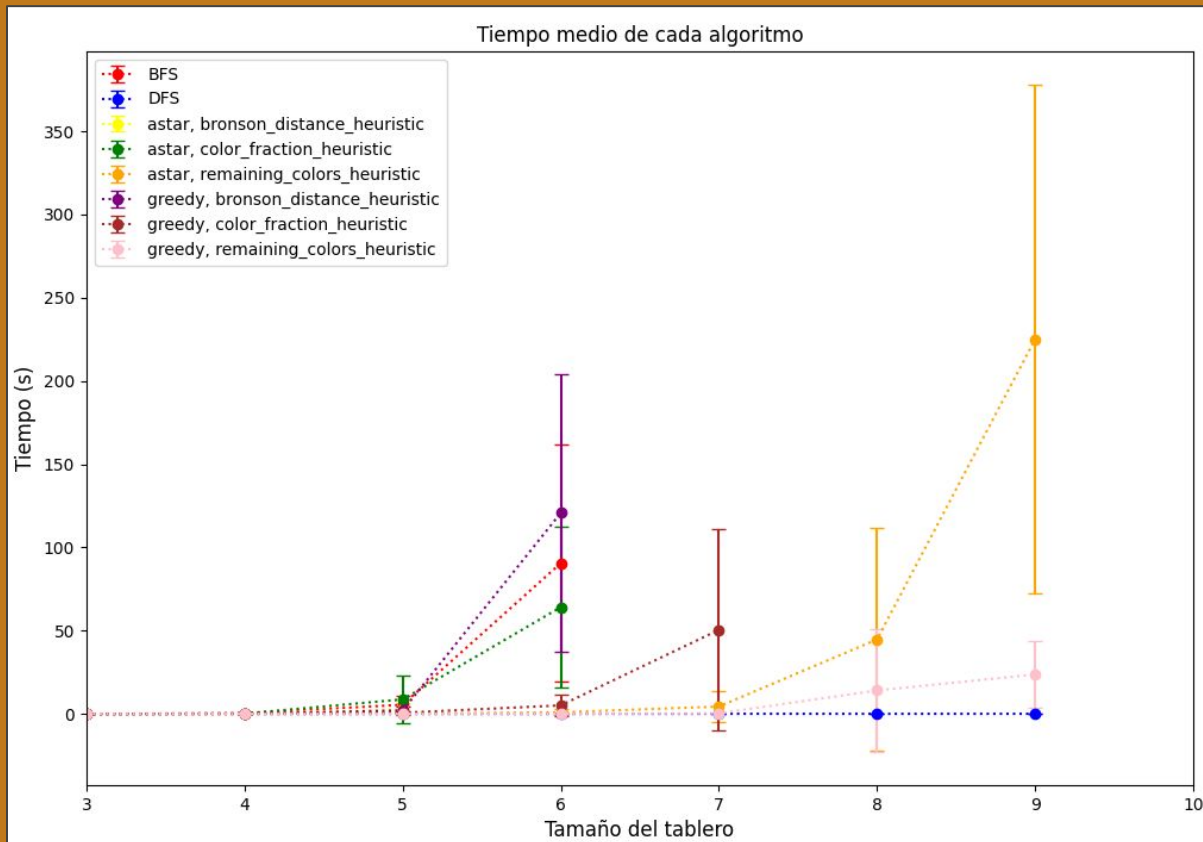
Para obtener el valor de la heurística se realiza un grafo con las celdas no controladas y por medio de *Dijkstra* se calcula el valor.

Es **admisible** ya que estima la mínima cantidad de movimientos necesaria para llegar a la celda inferior derecha, que siempre va a ser \leq a la cantidad necesaria para ganar el juego.

★			1	2	3
	1		1	2	3
	1	1	2	2	3
1	1	2	3	3	4
1	2	3	3	4	5
2	3	3	3	4	5

En el ejemplo: $H_3=5$

Resultados - Tiempo de ejecución



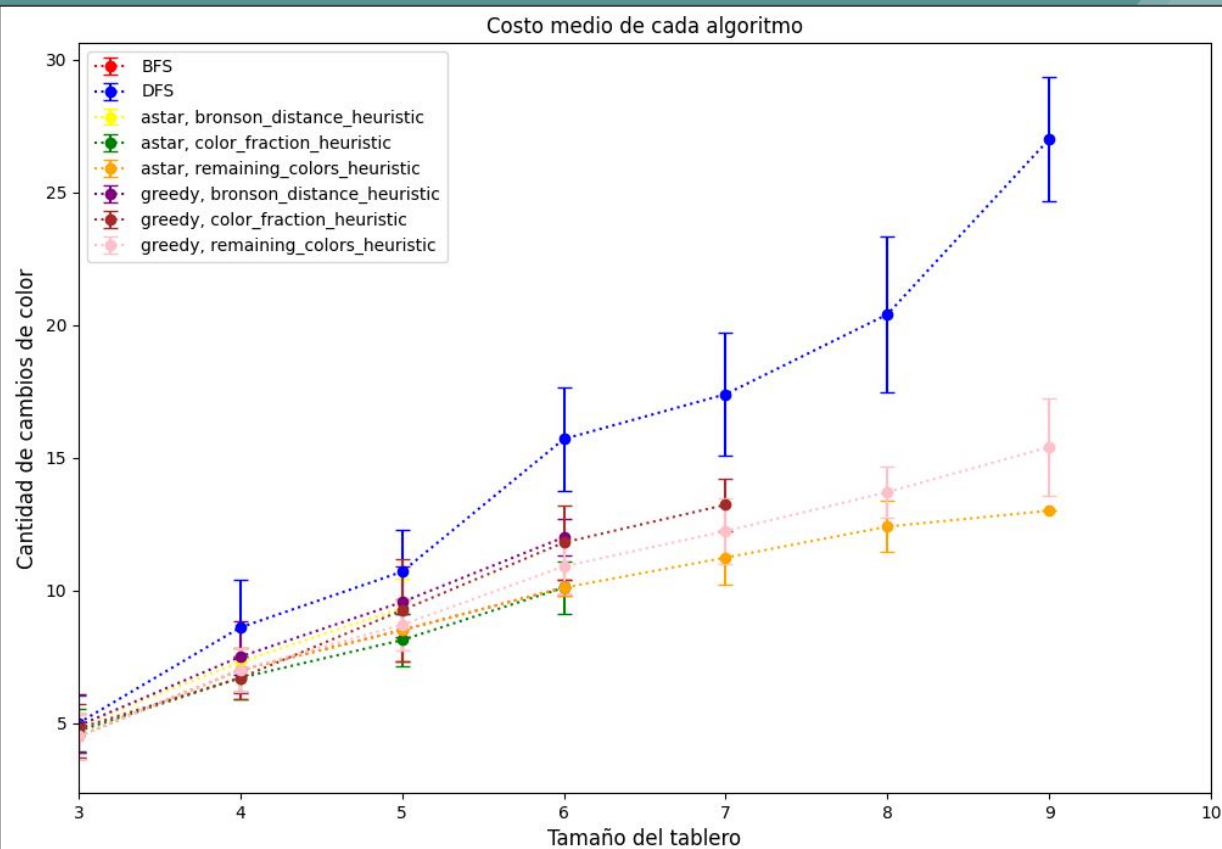
A partir de grillas de tamaño 7, los algoritmos:

- **BFS**
- **A*** con Color Fraction
- **A*** con Bronson
- **Greedy** con Bronson

demoraron demasiado como para analizarlos.

Se ve que **DFS** es el más rápido, mientras que tanto **A*** como **Greedy** terminan en un tiempo razonable en comparación al resto si se usa la Heurística de Cantidad de colores restantes.

Resultados - Costo de solución

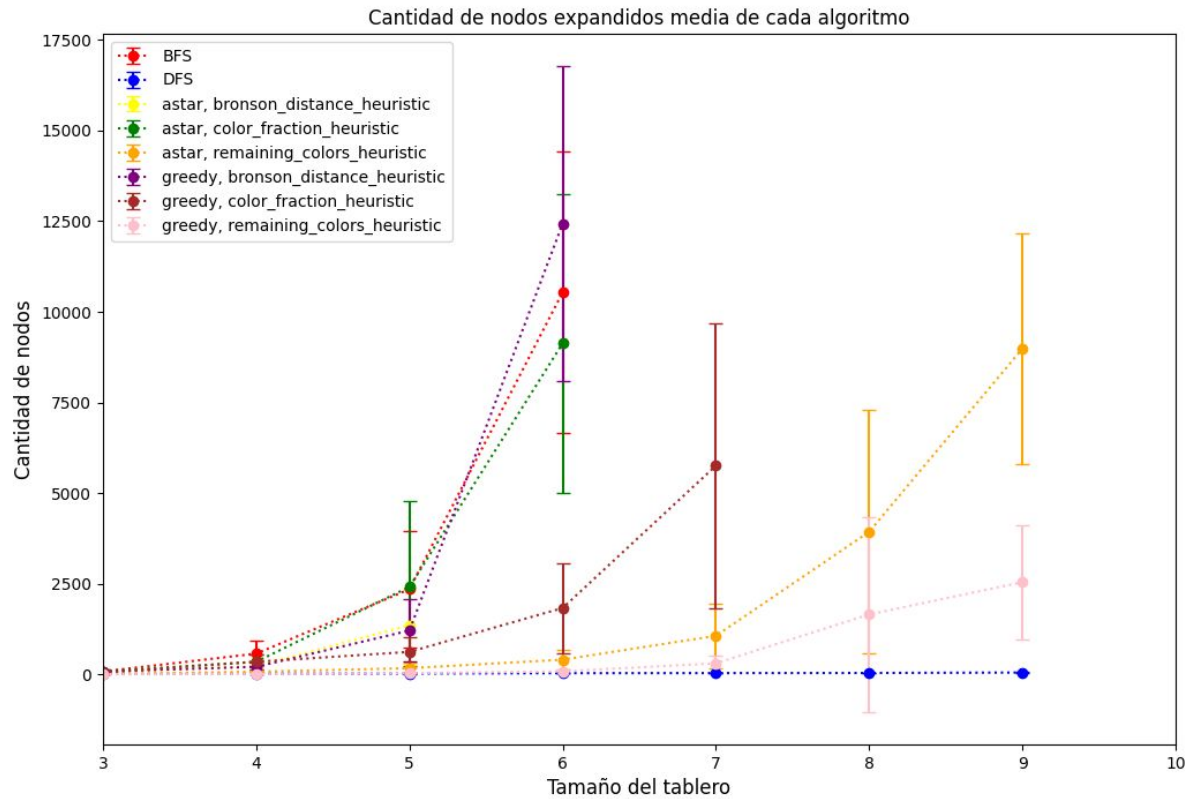


Se ve que **DFS** es el de mayor costo con diferencia sobre el resto, además de tener un error considerable por su modo de resolución.

El de menor costo fue **A*** con la heurística de **cantidad de colores restantes** como heurística, que además es el valor óptimo, siendo igual a BFS en tamaños pequeños.

Cerca tenemos a **Greedy** con la misma heurística, aunque se aprecia que su costo no es óptimo.

Resultados - Nodos expandidos

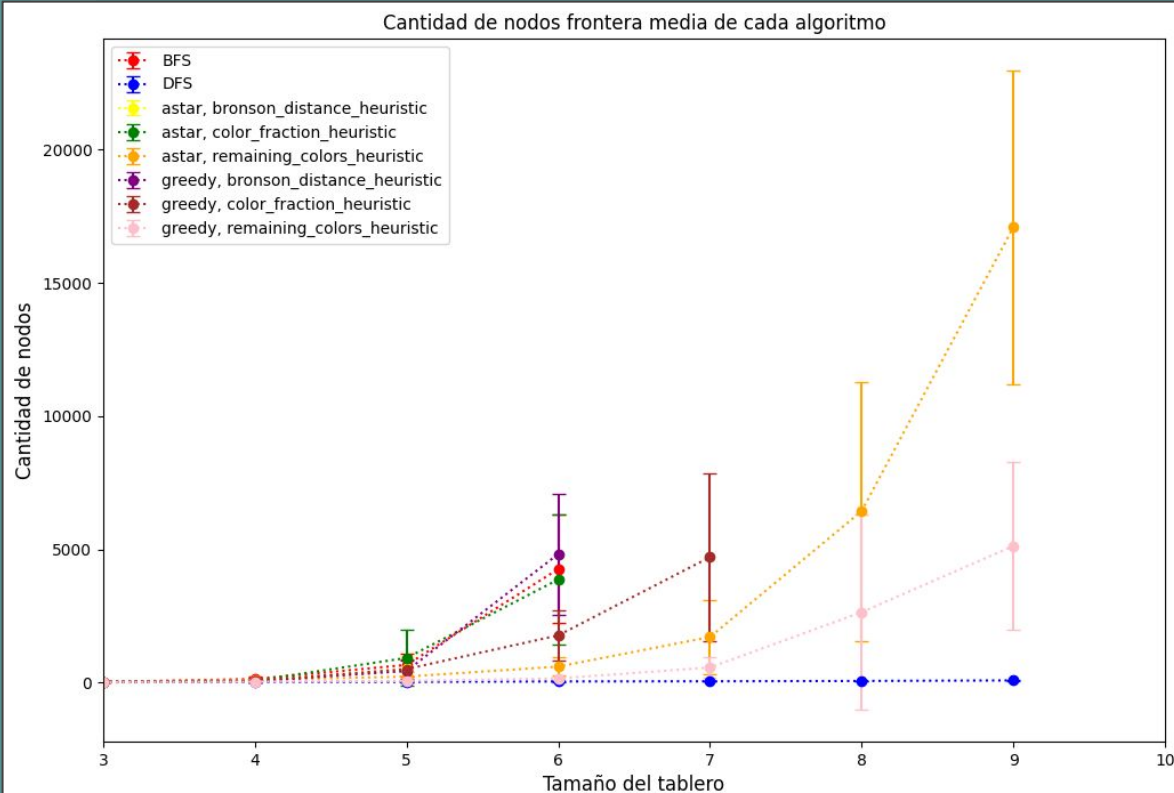


DFS es el algoritmo que menos nodos expande.

Notemos que **A*** con bronson se dejó de medir desde los tableros de 6x6 ya que el tiempo de ejecución se incrementó exponencialmente.

Si bien se puede observar que **Greedy** con Bronson es el que más nodos expande, seguido por **BFS**, pero si se dejara correr **A*** este debería expandirse más que Greedy e incluso más que **BFS**, dependiendo de la heurística.

Resultados - Nodos frontera



Se observa que el que más nodos tiene en la frontera al terminar es **A***, ya que, al considerar pasos anteriores de menor valor (costo + heurística) para “expandir” el árbol de soluciones, deja muchos nodos sin explorar. Luego está **Greedy** que también expande varios nodos, pero no al nivel de **A***.

En cambio **DFS** al siempre elegir el mismo camino no evalúa caminos alternativos, más aún cuando el número máximo de turnos es elevado.

Conclusiones sobre los resultados

- ❖ **BFS** es muy mala opción para obtener soluciones al no poder terminar en tiempos razonables para tamaños de grilla superiores a 7x7. Para obtener un valor óptimo **A*** incluso con una “mala” Heurística es mejor opción.
- ❖ **DFS** es una buena alternativa para comprobar rápidamente si existe una solución al problema, a menos que esta sea de un costo del mismo orden que la cantidad máxima de turnos.
- ❖ En general, es mejor opción usar métodos informados para encontrar una solución, aunque una “mala” Heurística puede hacer que demoren demasiado. En el caso de **A***, hasta incluso parecerse a **BFS**.

Método y Heurística a usar en el Fill-Zone

Método elegido: A* o Greedy

- ❖ A pesar de que **A*** devuelve una solución de costo óptimo, **Greedy** devuelve una solución de costo razonable en menor tiempo, por lo que dependiendo de la cantidad de turnos máxima del juego ambas pueden ser la mejor opción si el objetivo es simplemente ganar.

Heurística elegida: Cantidad de colores restantes

- ❖ Esta Heurística fue la única admisible que pudo terminar en tiempos razonables tanto con **A*** como con **Greedy**, mostrando su eficacia con tableros de hasta 9x9.
- ❖ Igualmente creemos que **Bronson** puede ser una mejor alternativa para tableros de dimensiones mucho mayores, pero para los probados demora demasiado en terminar como para considerarlo como opción.

¡Gracias!

¿Preguntas?