TP1 Métodos de Búsqueda

Sistemas de Inteligencia Artificial ITBA 2023 Q1

Grupo 4

Integrantes

Gaspar Budó Berra

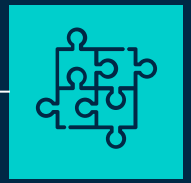
Marcos Dedeu

Marcus Galea Jacobsen

Santiago Hadad

Bruno Squillari

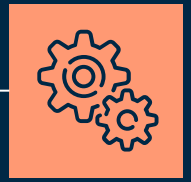
CONTENIDO



01

Ejercicio 1

Conclusiones sobre el 8-Puzzle



02

Ejercicio 2

Problema elegido, Estructura de estado, heurísticas y conclusiones

8 Puzzle Estado y Heurística

Estructura de estado: Matriz de 3x3, espacio vacío como null

Heurística 1: La suma de las distancias de cada tile a su posición objetivo (Distancia Manhattan)

Heurística 2: Cantidad de tiles que están fuera de su lugar.

8 puzzle - Método de Búsqueda

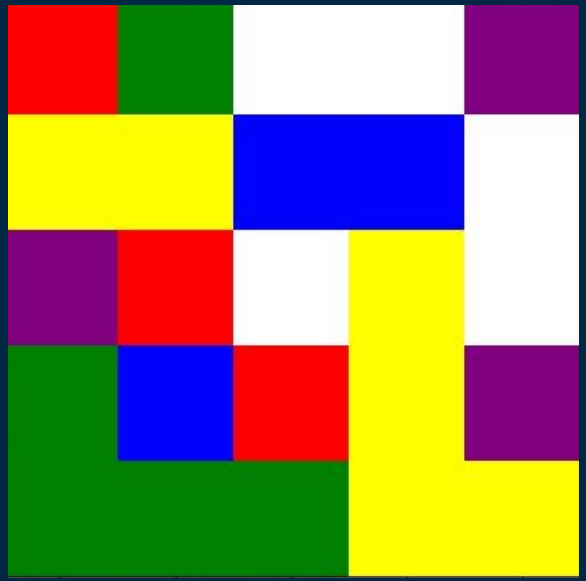
¿Método de búsqueda, con qué heurística y por qué?

- A* Search

- Distancia manhattan

Fill zone

- Se considera cada movimiento con costo uniforme igual a 1.
- Se busca la solución con menor cantidad de movimientos.
- Se puede variar la cantidad de colores y la dimensión de la matriz.
- Se usa un array de posiciones ya "conquistadas" y una matriz como estructura de estado.



Búsqueda de la solución

- Métodos desinformados:
  - DFS
  - BFS
- Métodos informados
  - A*
  - Greedy

Estudiamos el comportamiento de los diferentes algoritmos de búsqueda:

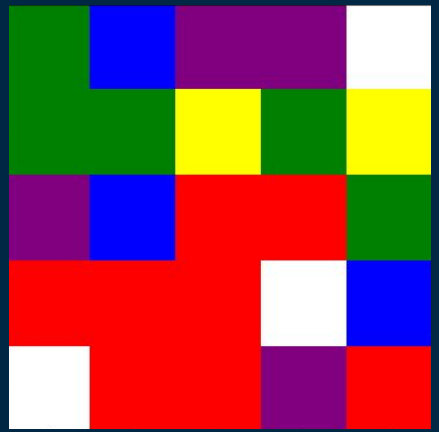
- Se varía la dimensión de la matriz, manteniendo la cantidad de colores constante.
- Dada una dimensión, todos los métodos se enfrentan a la misma matriz.
- También se generan matrices aleatorias de NxN, y se estudia el costo.

Comparación heurísticas: Admisible 1

Cantidad de colores en la matriz menos uno.

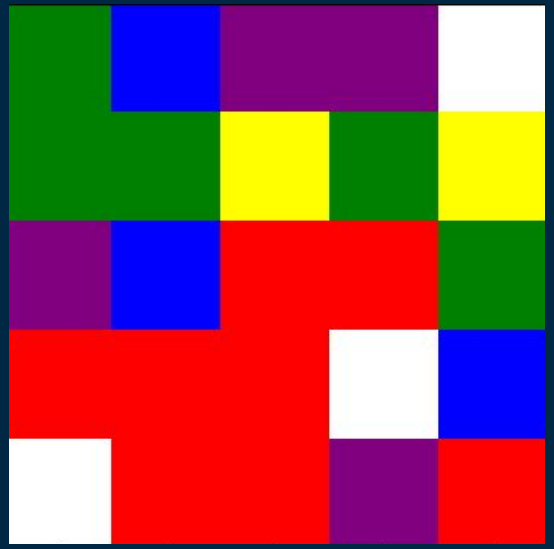
$$\sharp Colores \leq \sharp CostoTotal$$

Para cambiar un color hay que hacer un movimiento por lo menos.



Comparación heurísticas: Admisible 2

La cantidad de colores en la zona adyacente a la zona “conquistada”.

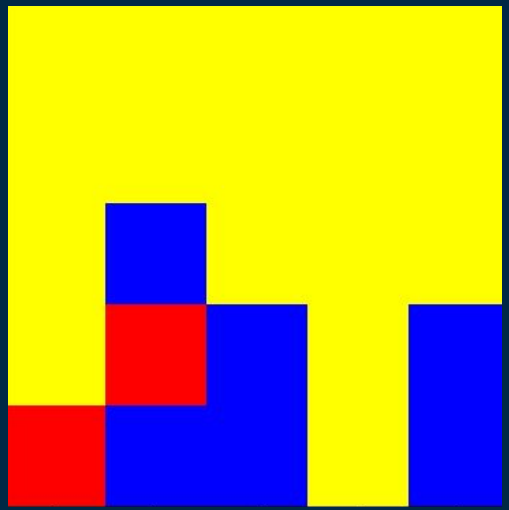


- La cantidad de colores en la zona adyacente es menor o igual a la cantidad de colores en la matriz menos uno.
- Esta heurística es dominada por la anterior.

$$h_2(n) \leq h_1(n); \forall n$$

Comparación heurísticas: No Admisible

La cantidad de celdas no “conquistadas” dividido por la cantidad de colores faltantes.



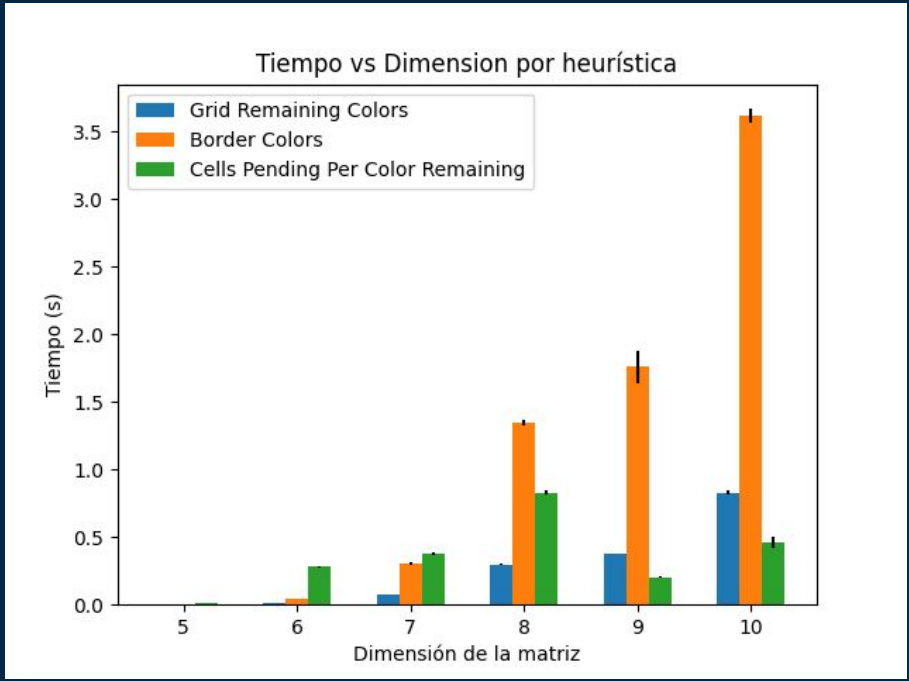
- Llegamos a esta heurística mediante la lógica de que el mejor camino sería aquel que deje menos celdas no conquistadas.
- Sin embargo, claramente puede verse en este ejemplo porque no es admisible: se puede llegar a la solución en 2 pasos, pero el resultado de la heurística será 8 (celdas) / 2 (colores) = 4

Comparación entre heurísticas: Tiempo

Se analiza el promedio de 10 ejecuciones del método A* con cada heurística para distintas dimensiones.

Condiciones:

- 6 colores
- Todas las ejecuciones de una dimensión utilizan la misma matriz

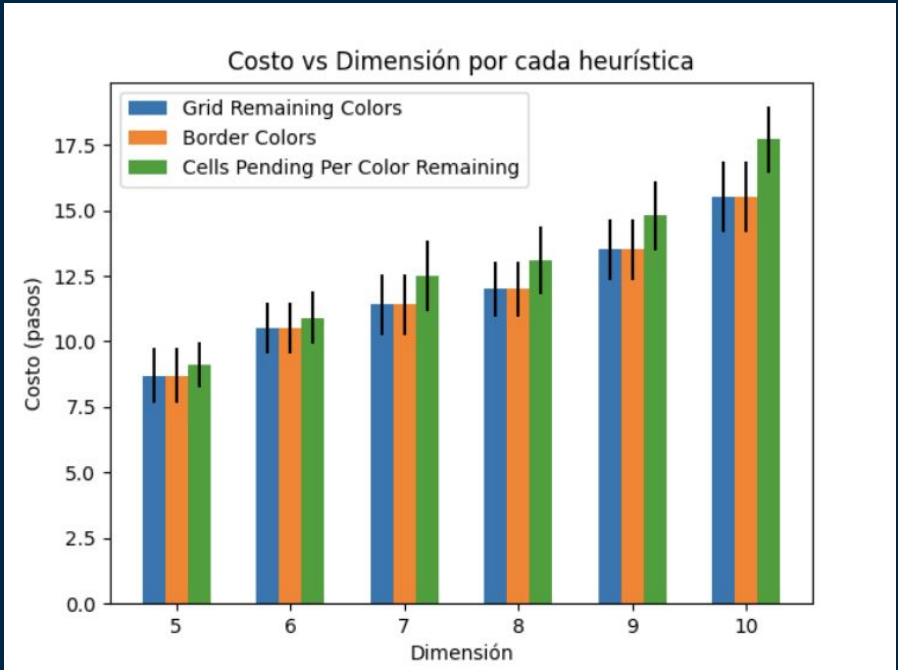


Comparación entre heurísticas: Costo

Se analiza el promedio de 10 ejecuciones del método A* con cada heurística para distintas dimensiones.

Condiciones:

- 6 colores
- Se utiliza una matriz aleatoria distinta en cada ejecución.

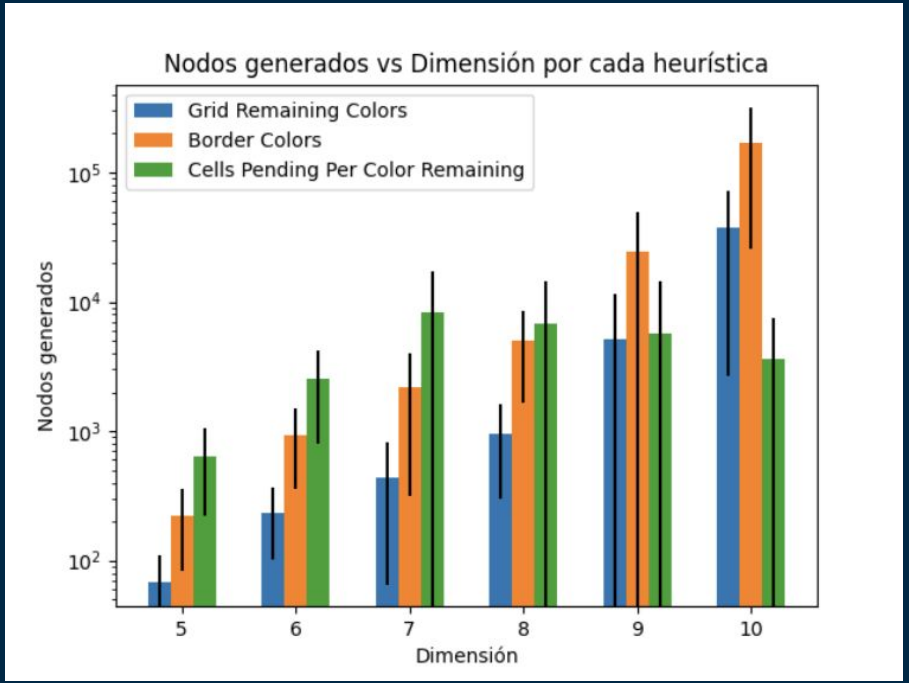


Comparación entre heurísticas: Nodos generados

Mismas condiciones que el punto anterior.

Datos:

- 6 colores
- Variamos la matriz base por lo que incluimos el desvío estándar del promedio de los datos.
- Hay una fuerte relación entre la cantidad de nodos generados y la dificultad de la matriz a resolver.

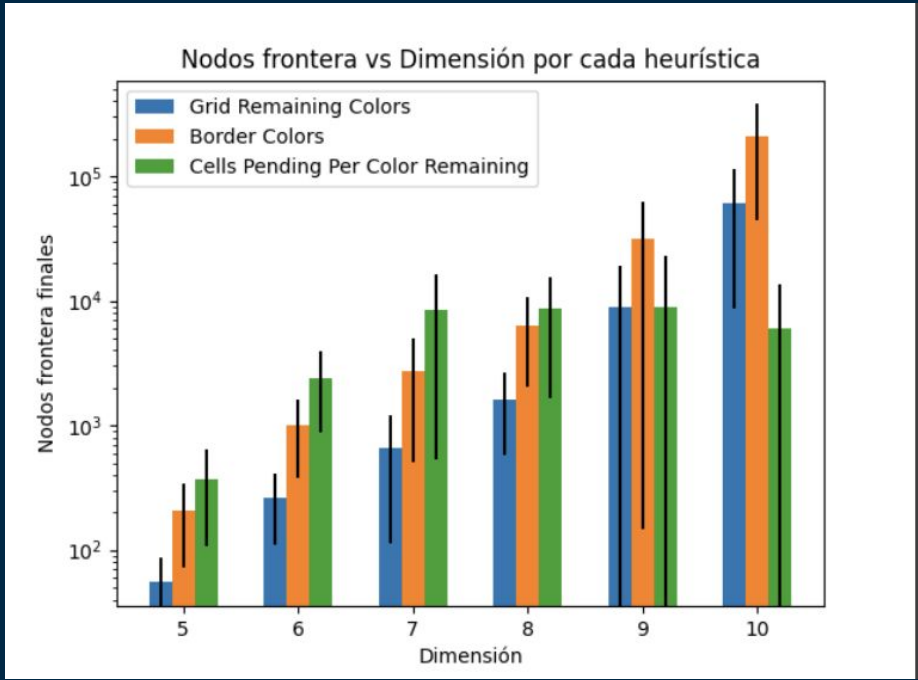


Comparación entre heurísticas: Nodos frontera

Mismas condiciones que el punto anterior.

Datos:

- 6 colores
- Variamos la matriz base por lo que incluimos el desvío estándar del promedio de los datos.
- También está íntimamente relacionado con la dificultad de la matriz a resolver.



Comparación entre heurísticas: Conclusiones

Comparación entre admisibles:

- La heurística que cuenta la cantidad de colores faltantes mostró una amplia superioridad en todos los aspectos respecto de la que cuenta los colores en el borde "conquistado".
- Como la primera heurística domina a la segunda, entonces expande menos nodos. Lo que lleva a que se ejecute en menor tiempo.

Comparación entre heurísticas: Conclusiones

Comparación entre la mejor admisible y no admisible:

Tiempo:

A mayor dimensión la heurística 1 tardaba menos de medio segundo más.

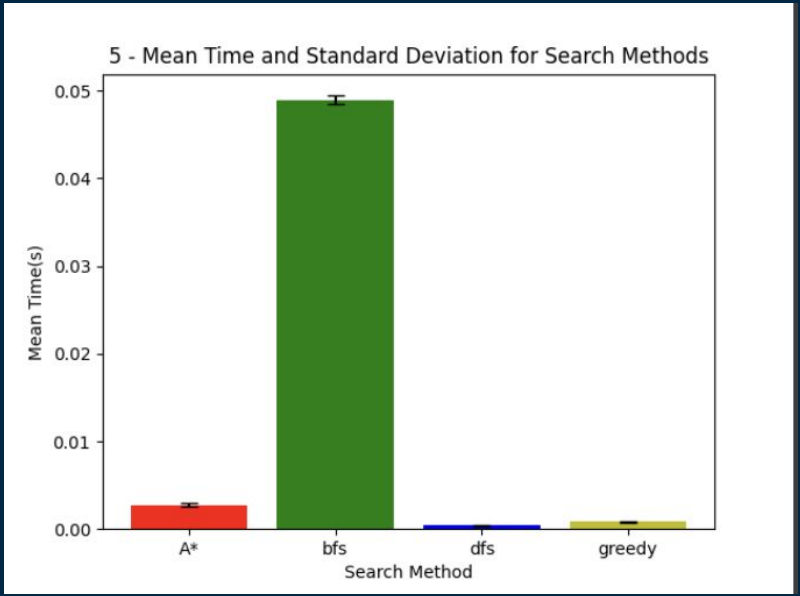
Costo:

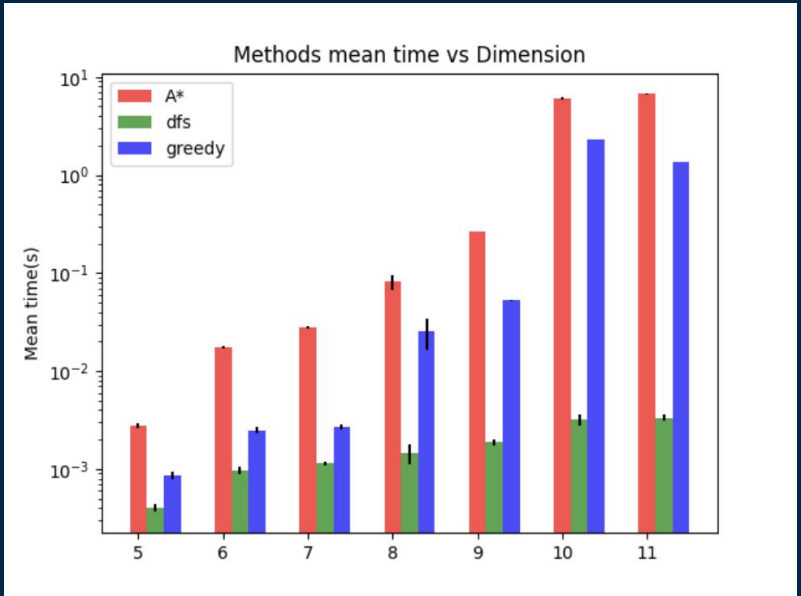
A mayor dimensión la no admisible genera un costo mayor de en promedio alrededor de 2 pasos.

- La heurística no admisible tiene un comportamiento favorable en ciertos tables que responden a un patrón.

Comparación de métodos: Tiempo

Se varía la dimensión, y se analiza la relación entre la misma y el tiempo para hallar la solución separando por método. (Siempre 6 colores)





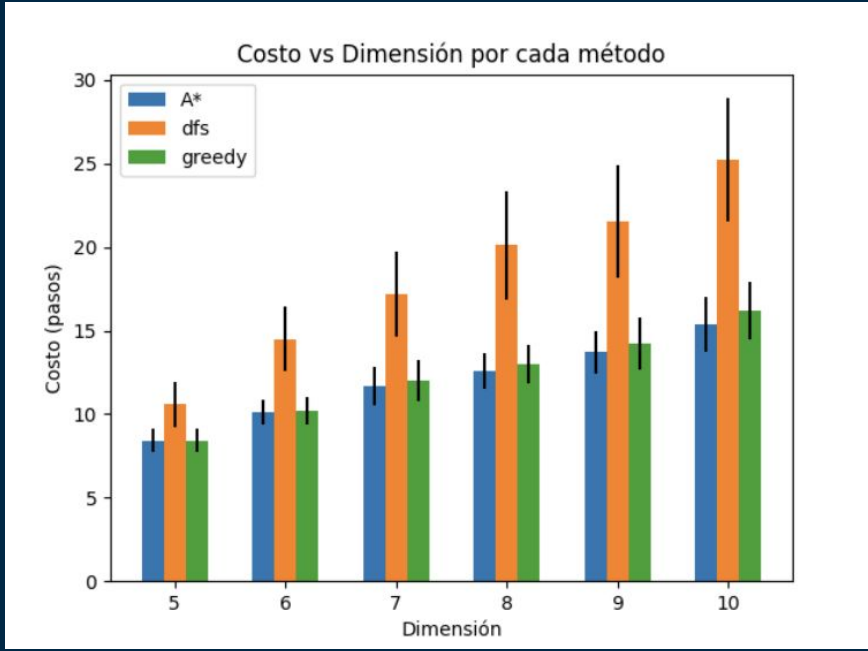
Comparación de métodos: Costo

Se varía la dimensión, y se analiza la relación entre la misma y el costo para hallar la solución separando por método.

No se muestra bfs por su diferencia exponencial, lo cual no aporta.

Condiciones:

- 6 colores
- Variamos la matriz base por lo que incluimos el desvío estándar del promedio de los datos.



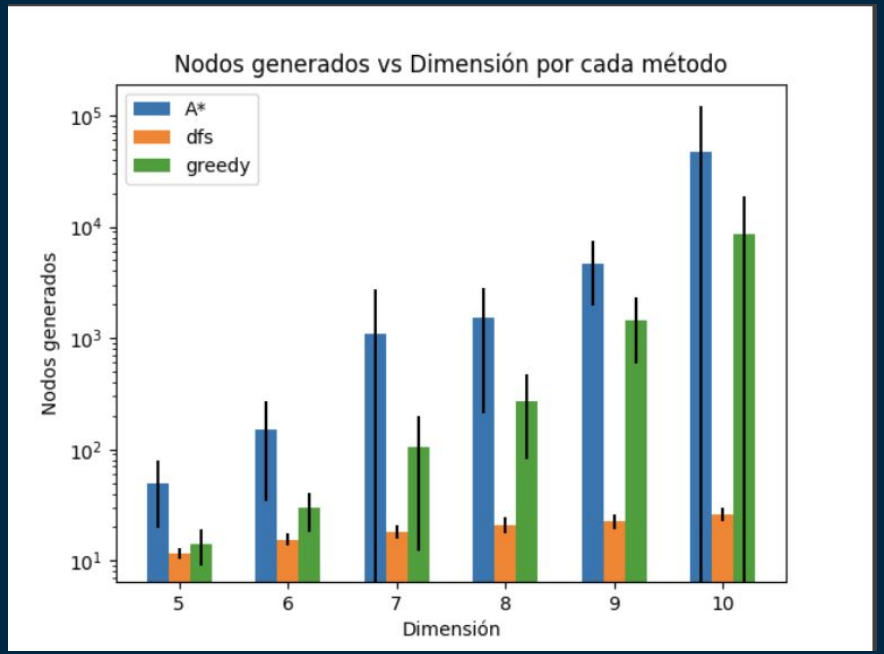
Comparación de métodos: Nodos generados

Se varía la dimensión, y se analiza la relación entre la misma y los nodos generados requeridos para hallar la solución separando por método.

Mismas condiciones que el punto anterior

Condiciones:

- 6 colores
- Variamos la matriz base por lo que incluimos el desvío estándar del promedio de los datos.



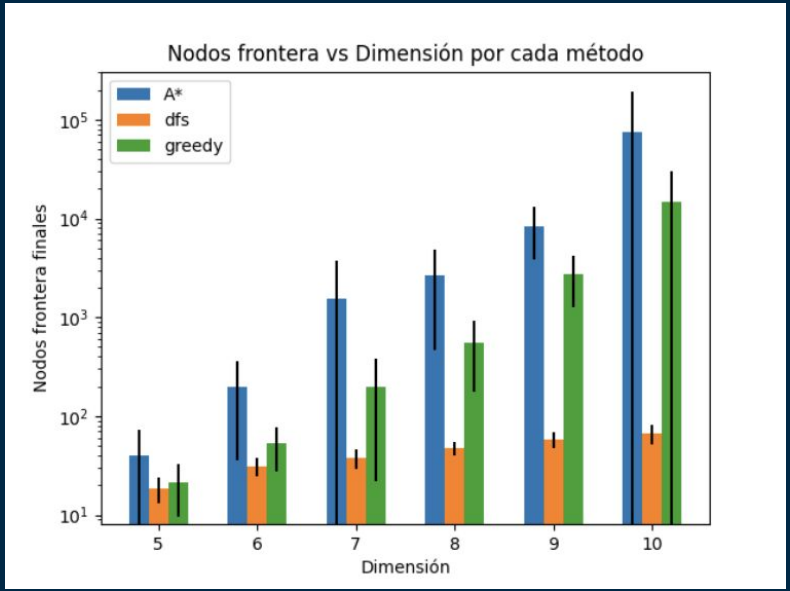
Comparación de métodos: Nodos frontera

Se varía la dimensión, y se analiza la relación entre la misma y los nodos frontera requeridos para hallar la solución separando por método.

Mismas condiciones que el punto anterior

Condiciones:

- 6 colores
- Variamos la matriz base por lo que incluimos el desvío estándar del promedio de los datos.



Comparación entre métodos: Conclusiones

- Los datos obtenidos son consistentes con lo esperado.
- A* siempre encuentra la solución óptima. Mientras más grande es el tablero, Greedy tiende a encontrar soluciones que no son óptimas.
- El costo de ejecución, la cantidad de nodos frontera y nodos generados son siempre mayores en el caso de A*.
- DFS encuentra una solución cualquiera de manera rápida.
- BFS al explorar todo el árbol sin ninguna heurística, en muchas ocasiones, no llega a finalizar su ejecución por la cantidad de memoria que necesita.

Solución GIF