ITBA
SIA
TP 1: Métodos de Búsqueda
No Informados e Informados

Grupo 2
José Torreguitar - 57519 - jtorreguitar@itba.edu.ar
Tomas Soracco - 56002 - tsoracco@itba.edu.ar
Sofía Picasso - 57700 - spicasso@itba.edu.ar

Introducción

El objetivo de este informe es describir y explicar el trabajo realizado, en el cual se analizó un problema dado por la cátedra. El mismo se representó debidamente y se implementaron distintos algoritmos de búsqueda (informados y no informados) para encontrar la solución a dicho problema.

La cátedra también proveyó una API con las interfaces Heuristic, State, Problem y Rule, la cual también modula el trabajo en dos partes: el modelo del problema dado y la implementación del motor que resuelve el problema. De esta forma, dicho motor puede ser utilizado para resolver cualquier problema que tenga la misma API.

El problema se resolvió utilizando distintos algoritmos de búsqueda y criterios heurísticos para los mismos, de manera tal que se pudiera comparar la performance de cada uno de ellos. Esto nos permitió llevar a cabo un análisis de la efectividad de cada uno de los algoritmos a la hora de resolver un problema, y cuales son las condiciones en las que convendría utilizar cada uno de los mismos.

Problema

El problema otorgado al grupo fue el de *simple squares,* el cual se juega en un tablero cuadrado. El objetivo del juego es empujar cuadrados de distintos colores a un punto del mismo color que se encuentra en el tablero. Los cuadrados sólo pueden moverse en una dirección, sin embargo, esta dirección puede ser cambiada si el cuadrado cae en un espacio especial con una flecha, la cual apunta hacia la nueva dirección a la que se dirige ahora el cuadrado.

El objetivo del juego es lograr que cada cuadrado llegue al círculo de su mismo color en la menor cantidad de movimientos. En nuestro caso, los cuadrados no pueden seguir moviéndose en la misma dirección cuando llegan a algún borde del tablero. Un cuadrado también puede empujar a otro si se encuentra en su camino. Esto cuenta como un solo movimiento, aunque dos o más bloques se mueven en ese movimiento.

Dificultad de los casos

Acordamos que la dificultad de resolución de un caso estaba relacionada a cuán desafiante es para la computadora resolver el mismo, lo cual nos trajo distintas perspectivas. Decidimos priorizar los parámetros de costo de la solución y cantidad de nodos expandidos para determinar la dificultad.

En cuanto a los algoritmos utilizados para realizar el análisis, decidimos utilizar todos los disponibles para cada caso. Esto nos permitió realizar una comparación objetiva de las ventajas y desventajas de cada uno en la práctica.

Función de costo

Se decidió utilizar una sola función de costo ya que se busca llegar al objetivo en la menor cantidad de movimientos. Por lo tanto, el costo aumenta en una unidad con cada movimiento realizado.

Heurísticas

Se implementaron tres heurísticas distintas para el problema dado, descritas a continuación:

HeuristicDistance

Esta heurística contempla la distancia de un cuadrado a su posición correcta. Puede parecer que esta heurística es admisible, pero si un cuadrado A puede acercar un cuadrado B al objetivo B empujándolo al realizar un movimiento en su dirección, se puede estar sobreestimando la misma. Por lo tanto no es admisible.

HeuristicInPlace

Para este caso se analiza la cantidad de cuadrados que están en la posición correcta. En este caso se subestima la cantidad de movimientos a realizar. Por lo tanto esta heurística es admisible.

HeuristicWeightedDistance

Similar a *HeuristicDistance*, esta heurística tiene en cuenta la dirección de movimiento del cuadrado. Todo movimiento en la dirección del mismo hacia el objetivo tendrá un peso menor que un movimiento que no esté en la dirección de este. Puede haber casos en los que se sobreestime el costo de alcanzar el objetivo, por lo que la misma no es admisible.

Se decidió agregar HeuristicDistance ya que esta, a diferencia de HereusticInPlace, cambia en cada turno y por lo tanto permite a los algoritmos tomar decisiones más informados aunque no sea admisible. HeuristicWeightedDistance se agregó para comparar métodos informados con una heurística inferior con métodos no informados, para probar que tanto influye una heurística en estos métodos de búsqueda.

Métricas

Las métricas que se toman para cada ejecución son:

- Altura
- Nodos frontera
- Nodos expandidos
- Tiempo
- Costo

Observaciones

A partir del análisis realizado, pudimos hacer las siguientes observaciones:

- DFS vs BFS:
  - BFS explota una mayor cantidad de nodos .
  - BFS utiliza mayor cantidad de memoria.
  - La cantidad de nodos explotados en BFS crece exponencialmente con el branching factor mientras que en DFS esta puede compararse a una función lineal de la profundidad promedio de las soluciones.
- IDDFS vs DFS vs BFS:
  - IDDFS expande considerablemente más nodos que los otros algoritmos desinformados
  - IDDFS tiende a ser más costoso en tiempo que sus contrapartes desinformadas, sin embargo, este aumento es despreciable si lo que se busca es optimizar memoria puesto que la cantidad de nodos en frontera y analizados es insignificante comparado con DFS y BFS.
- Greedy vs Algoritmos desinformados
  - Dependiendo de la heurística, Greedy suele tener menor cantidad de estados explotados
  - Con heurísticas admisibles o cuya sobreestimación del costo al objetivo no es demasiada, se puede observar que la pérdida en cuanto a tiempo no sobrepasa las ventajas obtenidas en cuanto a memoria.
  - La cantidad de nodos frontera y nodos explotados varía notablemente según la heurística elegida
- A* vs Greedy
  - La heurística afecta la cantidad de nodos frontera y nodos expandidos para ambos algoritmos
  - Greedy explota una menor cantidad de nodos para la misma heurística
  - Se puede apreciar entre greedy y A* la misma yuxtaposición que entre BFS y DFS. En la mayoría de los casos A* tiene más nodos explotados que greedy puesto que este último busca en profundidad mientras que A* busca a lo ancho.

Conclusiones

Al momento de hallar la solución existen distintos algoritmos utilizables. Al elegir una estrategia hay que tener en cuenta que métricas vamos a priorizar.

- Si se busca optimizar el tiempo de ejecución se puede observar que al no tener que reordenar la frontera y no tener que explotar más de una vez múltiples nodos, BFS y DFS superan a los otros algoritmos en este aspecto. En DFS, sin embargo, esto es dependiente del problema y de cuán numerosas sean las soluciones posibles a lo largo del árbol.

- A la hora de optimizar los recursos de almacenamiento la clara opción es IDDFS, sin embargo, si se busca una solución que comprometa en menor medida el tiempo de ejecución puede elegirse greedy con una heurística admisible o que no sobreestime demasiado el costo para llegar al objetivo

- La heurística de weighted distance, que sobreestima notablemente el costo a la solución, impacta severamente sobre los recursos utilizados (tanto temporales como de memoria) por ambos greedy y A*. Las otras dos heurísticas cuyas estimaciones son más cercanas a la realidad tienen un impacto aceptable en estos recursos.

Anexo

Gráficos:

Tabla 21 - Greedy - Nodos expandidos por heuristica

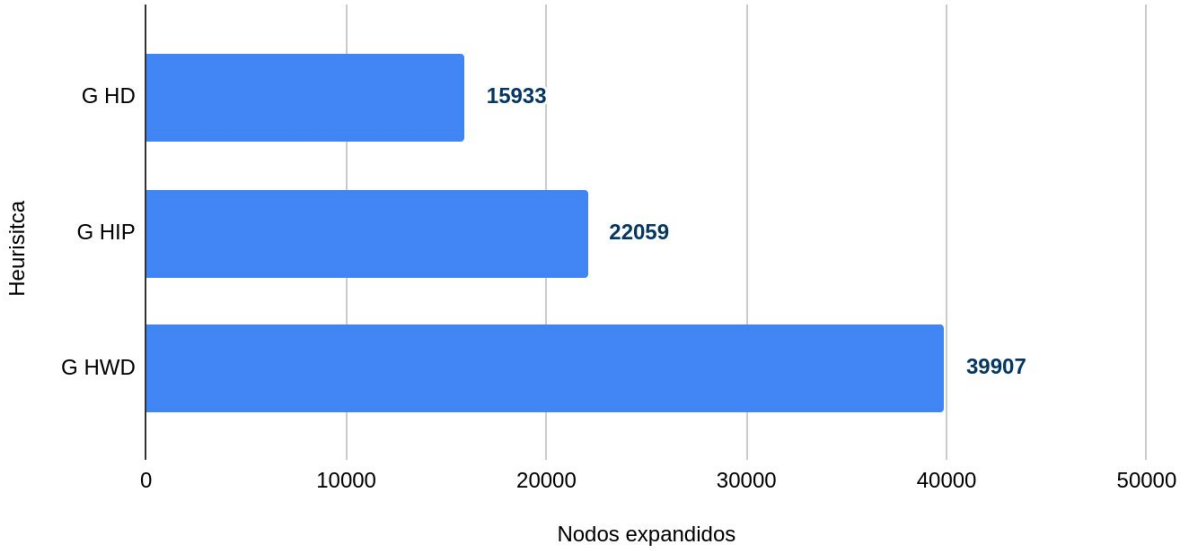
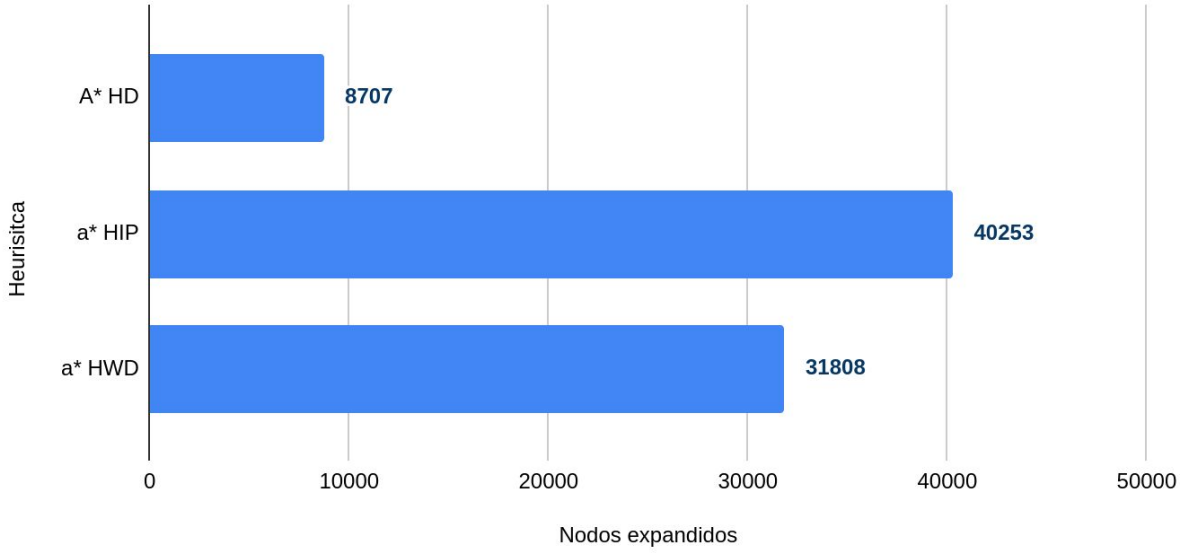
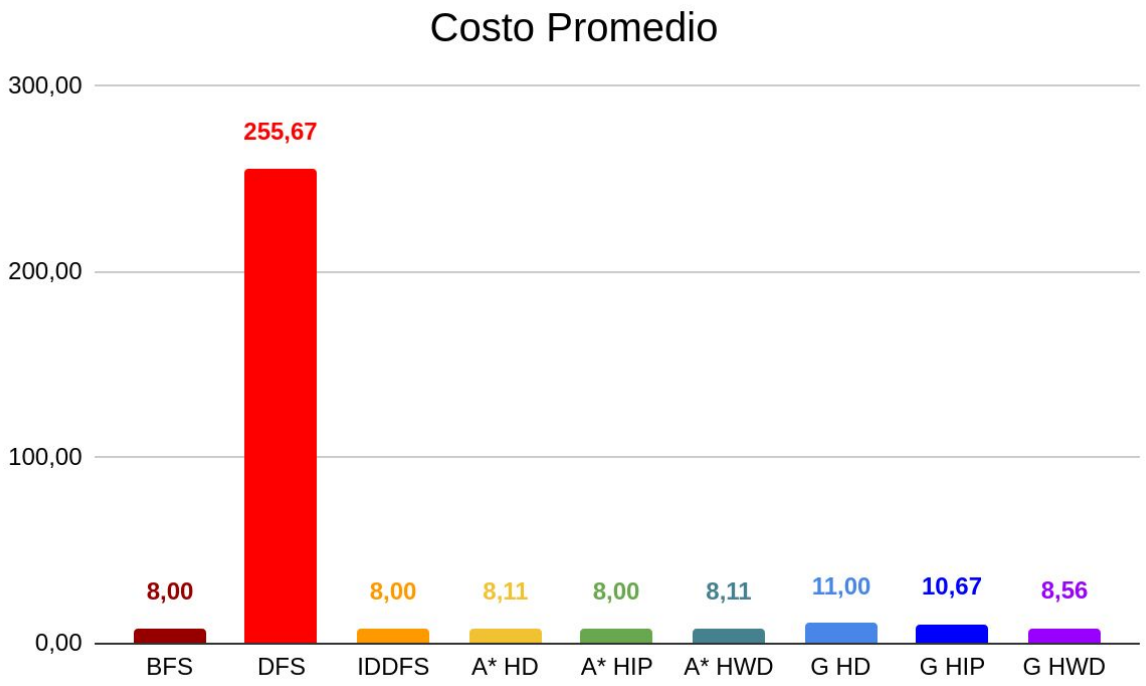
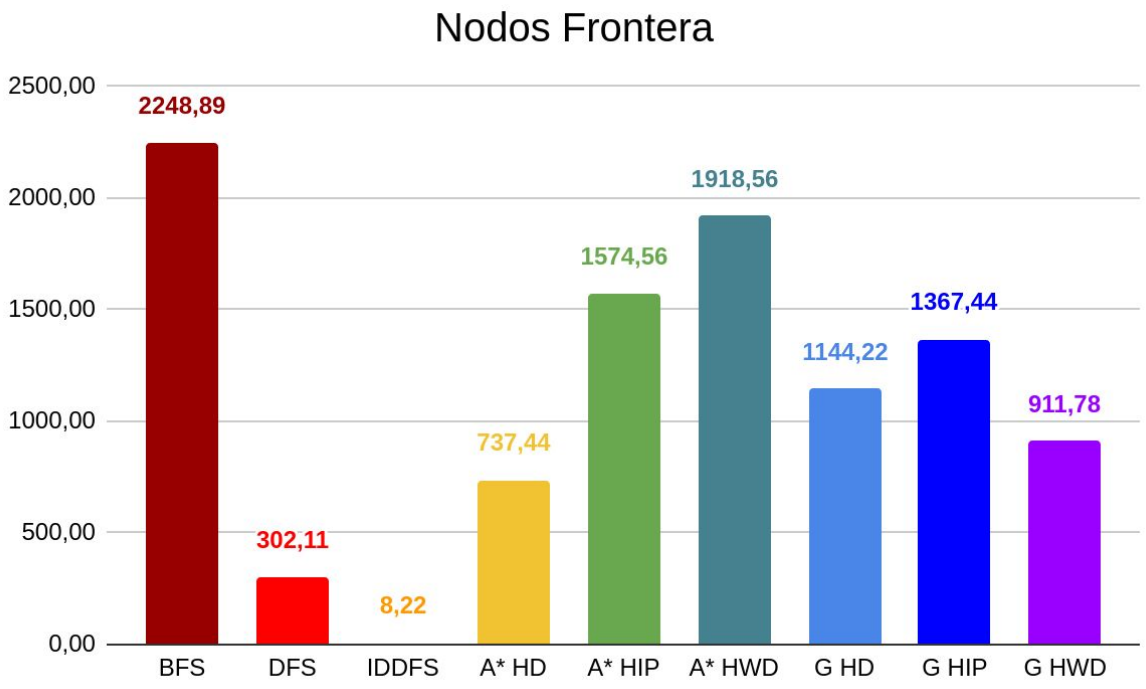
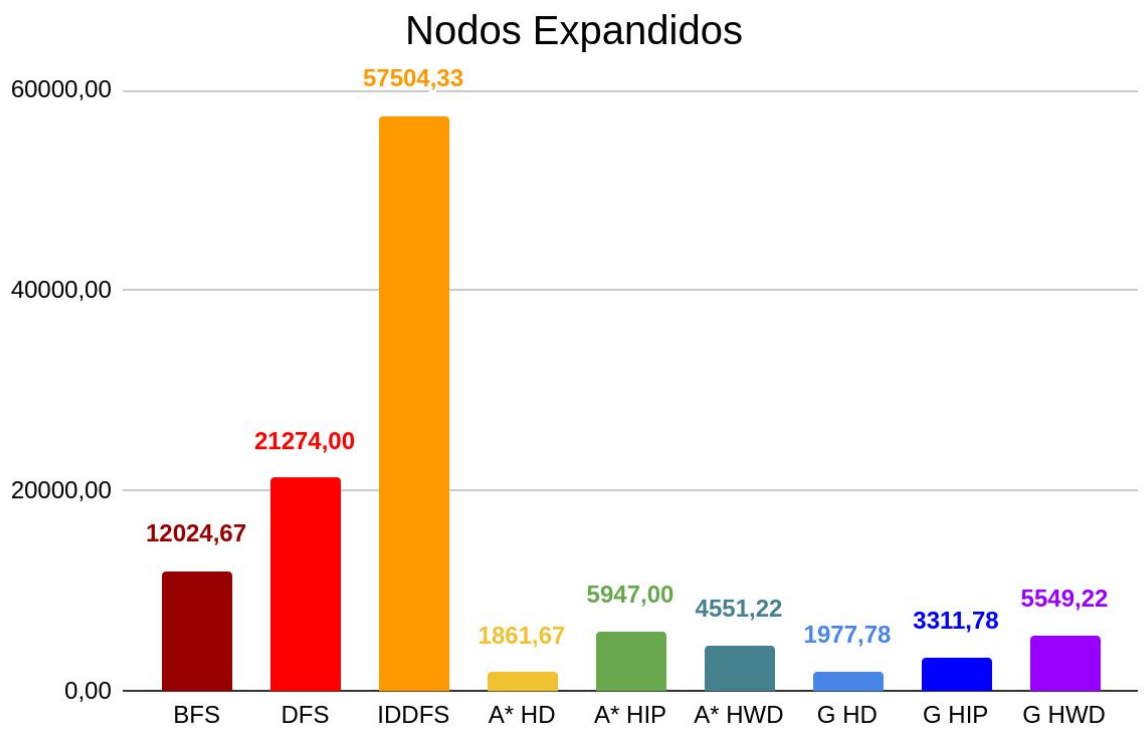


Tabla 21 - A* - Nodos expandidos por heuristica




Costo Promedio
300,00
200,00
100,00
0,00
255,67
8,00
8,00
8,11
8,00
8,11
11,00
10,67
8,56
BFS
DFS
IDDFS
A* HD
A* HIP
A* HWD
G HD
G HIP
G HWD


Nodos Frontera
2500,00
2000,00
1500,00
1000,00
500,00
0,00
2248,89
302,11
8,22
737,44
1574,56
1918,56
1144,22
1367,44
911,78
BFS
DFS
IDDFS
A* HD
A* HIP
A* HWD
G HD
G HIP
G HWD



Tablas utilizadas:

Board1	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	2	1	2	0,013	2
DFS	2	1	2	0,001	2
IDDFS	2	1	4	0	2
A* HD	2	1	2	0,004	2
a* HIP	2	1	2	0	2
a* HWD	2	1	2	0	2
G HD	2	1	2	0,001	2
G HIP	2	1	2	0	2
G HWD	2	1	2	0	2

Board2	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	2	2	3	0,001	2
DFS	2	2	2	0,001	2
IDDFS	2	2	5	0	2
A* HD	2	2	2	0,006	2
a* HIP	2	2	2	0	2
a* HWD	2	2	2	0	2
G HD	2	2	2	0,001	2
G HIP	2	2	2	0	2
G HWD	2	2	2	0	2

Board3	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	7	783	724	0,11	7
DFS	7	7	155	0,002	7
IDDFS	7	7	1221	0,116	7
A* HD	7	15	7	0	7
a* HIP	7	186	111	0,034	7
a* HWD	7	289	262	0,028	7
G HD	7	15	7	0	7
G HIP	7	15	7	0,001	7
G HWD	7	18	20	0	7

Board4	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	9	117	329	0,034	9
DFS	9	6	29	0,002	9
IDDFS	9	6	788	0,015	9
A* HD	9	7	9	0,001	9
a* HIP	9	87	154	0,008	9
a* HWD	9	43	262	0,004	9
G HD	9	7	9	0,001	9
G HIP	9	28	62	0,004	9
G HWD	9	8	326	0,003	9

Board5	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	5	153	117	0,031	5
DFS	5	7	174	0,002	5
IDDFS	5	7	186	0,004	5
A* HD	5	10	5	0,001	5
a* HIP	5	42	22	0,002	5
a* HWD	5	169	179	0,005	5
G HD	5	10	5	0,005	5
G HIP	5	88	92	0,001	5
G HWD	5	24	632	0,007	5

Board6	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	7	356	543	0,058	7
DFS	7	8	236	0,004	7
IDDFS	7	8	1026	0,013	7
A* HD	7	45	44	0,017	7
a* HIP	7	81	64	0,004	7
a* HWD	7	44	71	0,02	7
G HD	7	14	130	0,003	7
G HIP	7	34	74	0,001	7
G HWD	7	12	133	0,001	7

Board8	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	9	9	106	0,002	9
DFS	9	6	45	0	9
IDDFS	9	6	377	0,003	9
A* HD	9	29	64	0,002	9
a* HIP	9	11	92	0,001	9
a* HWD	9	12	52	0	9
G HD	9	10	85	0,003	9
G HIP	9	11	92	0	9
G HWD	9	8	73	0	9

Board20	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	27	1262	16816	0,420	27
DFS	43	49	62	0,080	43
IDDFS	27	31	110318	1,394	27
A* HD	31	2270	7915	0,585	31
a* HIP	27	1686	12823	1,224	27
a* HWD	27	2048	8323	0,532	27
G HD	49	336	1627	0,127	49
G HIP	38	1272	7416	0,449	38
G HWD	27	1578	8848	0,383	27

Board21	altura	nodos frontera	nodos expandidos	tiempo	costo
BFS	23	15443	89582	0,785	23
DFS	2252	2674	190761	1,872	2252
IDDFS	23	29	403614	4,171	23
A* HD	24	6452	8707	1,17	24
a* HIP	23	13406	40253	9,306	23
a* HWD	24	15114	31808	7,535	24
G HD	50	9846	15933	2,378	50
G HIP	47	10856	22059	1,87	47
G HWD	28	6555	39907	6,907	28